

VŠB – Technická univerzita Ostrava  
Fakulta elektrotechniky a informatiky

# **DIPLOMOVÁ PRÁCE**

2019

Bc. Vít Jandásek

VŠB – Technická univerzita Ostrava  
Fakulta elektrotechniky a informatiky  
Katedra kybernetiky a biomedicínského inženýrství

# **Optická kontrola elektronické součástky**

## **Optical Inspection of Electronic Parts**

# Zadání diplomové práce

Student: **Bc. Vít Jandásek**

Studijní program: N2649 Elektrotechnika

Studijní obor: 2612T041 Řídicí a informační systémy

Téma: **Optická kontrola elektronické součástky**  
**Optical Inspection of Electronic Parts**

Jazyk vypracování: čeština

## Zásady pro vypracování:

Při sériové výrobě průmyslových výrobků, tedy i elektronických součástek, je zapotřebí provádět kontrolu kvality hotového výrobku. Pro vizuální kontrolu se používá technologie nazvaná „strojové vidění“. Průmyslová kamera spolu s vhodným objektivem a osvětlením sejme obraz výrobku a přenesení jej do počítače. V počítači je pak nasnímaný obraz z kamery zpracován pomocí SW pro strojové vidění. Takový SW dokáže nasnímaný obraz vyhodnotit dle požadavků zadavatele. Diplomová práce se zabývá výběrem vhodných komponent testeru a zejména vývojem SW komponent pro výše popsaný typ testeru.

## Body zadání:

1. Rozbor problematiky zpracování obrazu v oblasti průmyslového testování.
2. Popis vývojových prostředí/technologií používaných pro návrh kamerových systémů - přehled a charakteristika.
3. Rozbor funkce a popis algoritmů strojového vidění pro požadované optické testy SMD součástky.
4. Teoretický rozbor a návrh řešení kamerové inspekce SMD součástky.
5. Návrh řízení již navržené mechanické koncepce testeru.
6. Návrh uživatelských rozhraní SW.
7. Návrh statistického zpracování výsledných dat.
8. Vývoj SW komponent v programovacím prostředí LabVIEW pro firemní testovací SW framework.
9. Nasazení a odladění vyvinutých komponent v průmyslové praxi.
10. Ověření funkce a zhodnocení.

## Seznam doporučené odborné literatury:

- [1] VLACH, Jaroslav, Josef HAVLÍČEK a Martin VLACH. *Začínáme s LabVIEW*. Praha: BEN - technická literatura, 2008, 247 s. ISBN 978-80-7300-245-9.
- [2] BRESS, Thomas J. *Effective labview programming*. 1st ed. Allendale: NTS Press, 2013, 701 s. ISBN 1934891088.
- [3] BITTER, Rick, Taqi MOHIUDDIN a Matt NAWROCKI. *LabView advanced programming techniques*. 2nd ed. Boca Raton: CRC Press, c2007, 499 s. ISBN 0849333253.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **doc. Ing. Petr Bilík, Ph.D.**

Datum zadání: 01.09.2018

Datum odevzdání: 30.04.2019



---

doc. Ing. Jiří Koziorek, Ph.D.  
*vedoucí katedry*

---

prof. Ing. Pavel Brandštetter, CSc.  
*děkan fakulty*

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 30. dubna 2019

.....  
  
Bc. Vít Jandásek

Souhlasím se zveřejněním této diplomové práce dle požadavků čl. 26, odst. 9 Studijního a zkušebního řádu pro studium v magisterských programech VŠB-TU Ostrava.

V Ostravě 30. dubna 2019



.....  
Ing. Michal Harhaj

Chtěl bych poděkovat své rodině a přítelkyni za podporu a zázemí při studiu, svému vedoucímu práce doc. Ing. Petru Bilíkovi, Ph.D a všem svým kolegům za spolupráci při práci na projektu.

## **Abstrakt**

Tato diplomová práce popisuje návrh a realizaci testeru pro vizuální kontrolu SMD součástek v průmyslové výrobě. Diplomová práce se zabývá pouze návrhem a realizací SW a výběrem části HW do mechanické konstrukce. Software aplikace je vytvořen v grafickém programovacím jazyku LabVIEW za použití knihoven NI, to vše při dodržení standardů pro integraci firemního testovacího SW frameworku. Řídící část aplikace je částečně implementována na čipu FPGA, který zajišťuje rychlé a přesné čítání a časování. Výsledný software je po řádném otestování nasazen na automatizovaném testeru pro vizuální kontrolu v průmyslové výrobě.

**Klíčová slova:** strojové vidění, vizuální inspekce, LabVIEW, zpracování obrazu, FPGA, SMD součástky

## **Abstract**

This diploma thesis describes design and realisation of a tester for visual control of SMD components in production. The thesis describes design and implementation of SW and choice of HW, which installed in a mechanical construction. SW of the application is completely created in LabVIEW graphical programming language using NI libraries, while being able to comply and to be integrated into corporate SW framework for testing. Controlling part of the application is partially implemented on FPGA, which ensures appropriate counting and timing. The complete software was after proper testing period put to use on automated tester for visual control in industry.

**Key Words:** machine vision, visual inspection, LabVIEW, image processing, FPGA, SMD components



# Obsah

<b>Seznam použitých zkratk a symbolů</b>	<b>11</b>
<b>Seznam obrázků</b>	<b>13</b>
<b>Seznam tabulek</b>	<b>15</b>
<b>Seznam rovnic</b>	<b>16</b>
<b>1 Úvod</b>	<b>17</b>
<b>2 Rozbor problematiky zpracování obrazu v oblasti průmyslového testování</b>	<b>18</b>
2.1 Kamery . . . . .	18
2.2 Objektivy . . . . .	21
2.3 Osvětlovače . . . . .	24
2.4 Doplnující součásti sestavy pro strojové vidění . . . . .	26
2.5 Expozice . . . . .	27
<b>3 Popis vývojových prostředí/technologií používaných pro návrh kamerových systémů – přehled a charakteristika</b>	<b>28</b>
3.1 Vision Builder for Automated Inspection . . . . .	28
3.2 LabVIEW . . . . .	29
3.3 OpenCV . . . . .	32
<b>4 Rozbor funkce a popis algoritmů strojového vidění pro požadované optické testy SMD součástky</b>	<b>33</b>
4.1 Segmentace . . . . .	33
4.2 Prahování . . . . .	33
4.3 Matematická morfologie . . . . .	34
4.3.1 Dilatace . . . . .	35
4.3.2 Eroze . . . . .	35
4.4 Detekce hran . . . . .	36
4.4.1 Algoritmus detekce hran aproximující derivaci obrazu za pomoci difference	36
4.4.2 Algoritmus detekce hran založený na průchodu druhé derivace nulou . . .	37
<b>5 Popis testeru</b>	<b>38</b>
<b>6 Teoretický rozbor a návrh řešení kamerové inspekce SMD součástky</b>	<b>40</b>
6.1 Popis hlavních požadavků zákazníka . . . . .	40
6.2 Teoretický rozbor kamerové inspekce . . . . .	41
6.2.1 Hardware . . . . .	41

6.2.2	Software . . . . .	42
6.3	Návrh řešení kamerové inspekce . . . . .	43
6.3.1	Hardware . . . . .	43
6.3.2	Software . . . . .	47
6.3.3	Popis algoritmu . . . . .	48
6.3.4	Nastavení algoritmu . . . . .	51
<b>7</b>	<b>Návrh řízení testeru</b>	<b>53</b>
7.1	Požadavky na řízení . . . . .	53
7.2	Návrh řízení testeru . . . . .	53
7.3	Realizace logiky řízení na kartě pro rychlé signály . . . . .	54
7.4	Realizace logiky řízení na kartě pro pomalé signály . . . . .	58
<b>8</b>	<b>Návrh statistického zpracování výsledných dat</b>	<b>59</b>
8.1	Provozní statistika . . . . .	59
8.2	Návrh limitů parametrů optické inspekce . . . . .	59
<b>9</b>	<b>Vývoj SW komponent v programovacím prostředí LabVIEW pro firemní testovací SW framework</b>	<b>61</b>
9.1	Požadavky na GUI . . . . .	61
9.2	Implementace GUI . . . . .	61
9.3	Vývoj v SW frameworku . . . . .	62
9.4	Hlavní jádro – core_main . . . . .	64
9.5	Jádro pro kameru – core_x_y . . . . .	65
9.6	Hlavní uživatelské rozhraní – gui_main . . . . .	65
9.7	Konfigurační GUI pro FPGA – plugin_fpga_config . . . . .	66
9.8	Konfigurační GUI pro nastavení limitů – plugin_config_core . . . . .	67
9.9	Statistiky výroby – plugin_statistics . . . . .	68
9.10	Servisní panel – service_panels_8237R.vi . . . . .	69
9.11	Buffer snímků – img_buffer.lvclass . . . . .	69
9.12	Popis vztahů jednotlivých komponent aplikace v runtime . . . . .	71
<b>10</b>	<b>Nasazení a odladění vyvinutých komponent v průmyslové praxi</b>	<b>72</b>
10.1	Nastavení PC, instalace SW . . . . .	72
10.2	Testování . . . . .	72
<b>11</b>	<b>Závěr</b>	<b>74</b>
	<b>Literatura</b>	<b>75</b>
	<b>Přílohy</b>	<b>77</b>

## Seznam použitých zkratk a symbolů

CCD	– Charge-Coupled Device – typ obrazového snímače
CMOS	– Complementary Metal-Oxide Semiconductor – typ obrazového snímače
CPU	– Central Processing Unit – centrální procesorová jednotka
DC	– Direct Current – stejnosměrný proud
DIO	– Digital Input Output – digitální vstupně-výstupní
DPS	– deska plošných spojů
DUT	– Device Under Test – zkoušené zařízení
FFT	– Fast Fourier Transform – rychlá Fourierova transformace
FGV	– Functional Global Variable – funkcionální globální proměnná, prostor pro sdílená data v LabVIEW
FIFO	– First In First Out – typ paměti se sekvenčním přístupem, nejstarší prvek je vyčten jako první
FPS	– Frames per Second – snímků za sekundu
GUI	– Graphical User Interface - grafické uživatelské rozhraní
HW	– Hardware
FPGA	– Field Programmable Gate Array – programovatelné hradlové pole
GigE	– Gigabit Ethernet
ID	– Identification Data – identifikátor
IDE	– Integrated Development Environment – vývojové prostředí
IR	– Infrared – infračervený
lp/mm	– line pair per milimeter – párů linek na milimetr
MAX	– Measurement & Automation Explorer
MB	– megabajt
mm	– milimetr
MP	– megapixel
MTF	– Modular Transfer Function – funkce přenosu modulace
MWIR	– Mid-Wavelength Infrared – středněvlnné IR záření
NIR	– Near Infrared – blízké IR záření
LabVIEW	– Laboratory Virtual Instrument Engineering Workbench
LED	– Light Emitting Diode – světlo emitující dioda
LWIR	– Long-Wavelength Infrared – dlouhovlnné IR záření
OCR	– Optical Character Recognition – optické rozpoznávání znaků
PC	– Personal Computer – osobní počítač
PoE	– Power over Ethernet – napájení po Ethernetu
PWM	– Pulse Width Modulation – pulzně šířková modulace
ROI	– Region of Interest – oblast zájmu (v obraze)

SMD	– Surface Mount Device - součástka pro povrchovou montáž
subVI	– podprogram VI
SW	– Software
SWIR	– Short-Wavelength Infrared – krátkovlnné IR záření
USB	– Universal Serial Bus
UV	– Ultraviolet – ultrafialový
VBAI	– Vision Builder for Automated Inspection
VDM	– Vision Development Module
VI	– Virtual Instrument – virtuální instrument, souborový formát Lab-VIEW

## Seznam obrázků

1	Blokové schéma řetězce pro strojové vidění . . . . .	18
2	Příklad osvědčené průmyslové kamery – Basler ace . . . . .	19
3	Schematické porovnání CCD a CMOS . . . . .	20
4	Příklad masky na senzoru hyperspektrální kamery . . . . .	21
5	Běžný megapixelový objektiv s ohniskovou vzdáleností 16 mm . . . . .	22
6	Příklad soudkového a poduškového zkreslení objektivu . . . . .	24
7	Příklad různých typů osvětlovačů . . . . .	24
8	Schematický popis možných umístění osvětlovače . . . . .	25
9	Konfigurační rozhraní VBAI . . . . .	29
10	Measurement & Automation Explorer – parametry kamery . . . . .	30
11	Příklad čelního panelu a blokového diagramu . . . . .	30
12	Paleta funkcí „Machine Vision“ z modulu VDM v LabVIEW . . . . .	31
13	Příklad strukturních elementů . . . . .	34
14	Příklad morfologických operací . . . . .	35
15	Reprezentace hran v 1D obrazové funkci . . . . .	37
16	Diagram popisující průjezd součástky strojem. . . . .	38
17	Zjednodušené schéma stroje . . . . .	39
18	Snímky kontrolovaných součástek . . . . .	40
19	Příklady kontrolovaných vad součástek . . . . .	41
20	Příklady snímků z testování HW . . . . .	44
21	Snímek zvoleného osvětlovače a kontroléru . . . . .	45
22	Kamera Basler ace GigE . . . . .	46
23	Diagram popisující průběh zpracování a vyhodnocení obrazu . . . . .	48
24	Příklad binárního snímku součástky . . . . .	49
25	Lokalizovaná součástka . . . . .	50
26	Diagram popisující logiku na FPGA pro rychlé signály. . . . .	55
27	Stavový diagram bloku trigger kamery . . . . .	56
28	Stavový diagram bloku pulz ventilu . . . . .	58
29	Firemní testovací framework . . . . .	62
30	Užité SW komponenty . . . . .	63
31	Hlavní uživatelské rozhraní . . . . .	65
32	Konfigurační GUI pro FPGA . . . . .	66
33	Konfigurační GUI pro nastavení limitů . . . . .	67
34	GUI pro zobrazení statistik . . . . .	68
35	Servisní panel aplikace . . . . .	69
36	Diagram třídy img_buffer . . . . .	70
37	Komunikace mezi jednotlivými SW komponentami v inspekčním módu . . . . .	71

38	Komunikace mezi jednotlivými SW komponentami v konfiguračním módu . . . .	71
----	---	----

## Seznam tabulek

1	Základní parametry zvoleného osvětlovače . . . . .	45
2	Základní parametry zvolené kamery . . . . .	46
3	Základní parametry zvoleného objektivu . . . . .	47
4	Kontrolované parametry . . . . .	52

## Seznam rovnic

1	Kompletní segmentace obrazu . . . . .	33
2	Rovnice pro globální prahování obrazu . . . . .	33
3	Globální prahování . . . . .	33
4	Lokální prahování . . . . .	34
5	Modifikované globální prahování pro množinu šedých hodnot . . . . .	34
6	Duální transformace . . . . .	35
7	Dilatace . . . . .	35
8	Vlastnosti dilatace . . . . .	35
9	Eroze . . . . .	35
10	Vlastnosti eroze . . . . .	36
11	Robertsův operátor . . . . .	36
12	Laplaceův operátor . . . . .	36
13	Laplaceův operátor se zesíleným středem . . . . .	36
14	První dva Prewittovy operátory . . . . .	37
15	Výpočet expozičního času kamery . . . . .	42
16	Výpočet horního limitu parametru . . . . .	60
17	Výpočet dolního limitu parametru . . . . .	60



# 1 Úvod

Současná sériová průmyslová výroba klade mimo jiné velký důraz na rychlost či kvalitu za cenu snižování nákladů. Tyto zvýšené požadavky na kvalitu výrobků vedou mimo jiné na vizuální kontrolu výrobků. V souvislosti s rozvojem elektroniky a technologických možností jsou tyto vizuální kontroly realizovány ve výrobě stále častěji formou automatizované kamerové kontroly. Kromě nahrazení lidského faktoru v rámci kontroly lze nově zavést kontrolu i tam, kde kontrola dříve vůbec prováděna nebyla, protože by byla pouze lidskou silou nemožná. Obecně se tato problematika nazývá anglicky „Machine Vision“, česky strojové vidění nebo průmyslové zpracování obrazu. [1]

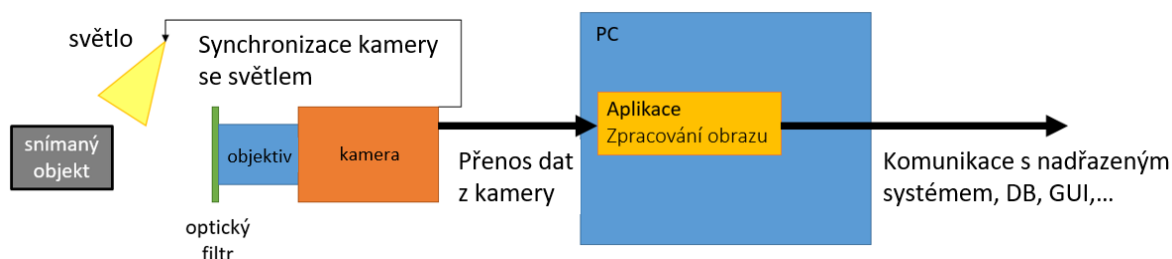
Sestava strojového vidění se skládá z kamery v průmyslovém provedení s vhodným objektivem, z průmyslového osvětlovače a z nadřazeného systému, typicky PC. Obraz je z kamery zasílán přes vhodné komunikační rozhraní do PC, kde je následně zpracován, analyzován a vyhodnocen. Výsledek vyhodnocení určuje, zda zobrazený výrobek odpovídá stanoveným požadavkům (rozměry, barva, čtení znaků, kódů atp.).

Teoretická část práce se zabývá rozbořem problematiky strojového vidění, popisem hardware užívaného pro kamerové inspekce, jejich typy a nejdůležitějšími parametry. Dále jsou popsána některá vývojová prostředí a vybrané algoritmy pro zpracování obrazu.

Praktická část práce se zabývá částečně výběrem hardware pro kamerovou kontrolu, dále návrhem, realizací a testováním software pro automatizovaný tester pro optickou kontrolu SMD součástek. Úkolem stroje je třídit bezvadné a vadné SMD součástky (dle požadavků zákazníka) na základě výsledku určeného při zpracování obrazu. Software zpracovává a vyhodnocuje obraz, provádí software také řízení stroje za pomoci 2 DIO karet s FPGA, vše s ohledem na požadavky zákazníka. Celý software je implementován v grafickém programovacím jazyce LabVIEW. Tento automatizovaný tester je nasazen v průmyslové výrobě. Závěr shrnuje dosažené výsledky z průběhu realizace a testování a získané zkušenosti.

## 2 Rozbor problematiky zpracování obrazu v oblasti průmyslového testování

Zjednodušeně lze úlohu strojového vidění popsat následovně: průmyslová kamera s vhodným objektivem, průmyslovým osvětlovačem a dalšími optickými součástmi (např. filtry) sejme obraz kontrolovaného výrobku. Obraz je z kamery zaslán po komunikačním rozhraní (GigE, USB 3.0) do nadřazeného systému (většinou PC), kde probíhá jeho zpracování (např. filtrace, prahování, změna jasu/kontrastu) a vyhodnocení (rozpoznání objektů, hledání hran, měření rozměrů, jasu, čtení čárových nebo maticových kódů, textu atp.), viz obr. 1.



Obr. 1: Blokové schéma řetězce pro strojové vidění

### 2.1 Kamery

Úlohou kamery v kamerovém systému je provést sejmутí obrazu se zadanými parametry. Kamery v úpravě pro průmysl se vyznačují svou jednoduchostí, kompaktností, odolností a dlouhou životností. Příklad vzhledu kamery je na obr. 2. Rozlišujeme několik základních parametrů kamer:

#### Komunikační rozhraní kamer

V dřívějších dobách byly k dispozici kamery jak s digitálním, tak s analogovým výstupem. Analogové kamery byly v počítači zapojené do analogových framegrabberů, což je zařízení pro zachytávání a digitalizaci obrazu. V dnešní době se pro nové aplikace využívají výhradně digitální kamery. Využívána jsou zejména následující rozhraní:

- **GigE Vision** – průmyslový standard vyvinutý na základě běžného protokolu gigabitového Ethernetu. Umožňuje spolehlivý přenos obrazových dat za pomoci standardní kabeláže a síťových prvků (které mají podporu pro gigabitový Ethernet) na relativně dlouhé vzdálenosti (až 100 metrů). Lze využít též PoE pro napájení kamery. Přenosová rychlost až 125 MB/s.
- **USB3 Vision** – standard založený na rozhraní USB 3.0. Kromě komunikace většinou zajišťuje i napájení kamery. Oproti GigE Vision dovoluje dosáhnout většího datového toku (až 350 MB/s) za cenu kratší maximální délky kabeláže (maximálně 5 m s pasivní kabeláží). Přenosová rychlost až 350 MB/s. [2]

- **Camera Link** – průmyslový standard pro nejnáročnější real-time aplikace s přenosovou rychlostí až 850 MB/s (v případě novějšího standardu HS teoreticky až 16 000 MB/s). Má poměrně rozměrnou a drahou kabeláž, v PC vyžaduje tzv. framegrabber s tímto rozhraním. Framegrabber je zařízení pro zpracování proudu obrazových dat z kamery, tato data poté mohou být dále ukládána, upravována, zobrazována apod. Využívá se v aplikacích, které vyžadují vysoký datový tok, tj. zejména pro vysokorychlostní a řádkové kamery či kamery s vysokým rozlišením. [2], [3]

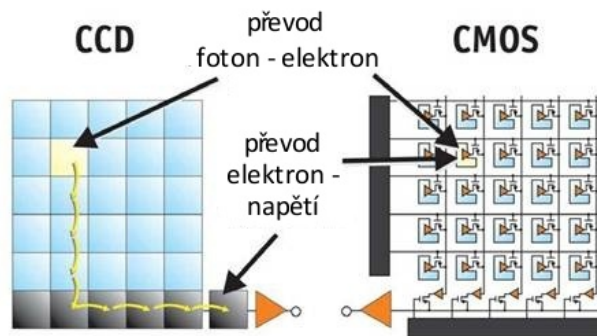


Obr. 2: Příklad osvědčené průmyslové kamery – Basler ace

### Typ a rozměry senzoru

Dle technologie výroby senzorů a způsobu přetváření světla na elektrický signál rozlišujeme dva druhy obrazových snímačů (viz obr. 3):

- **CCD** – světlocitlivá buňka převede dopadající foton na elektrický náboj, ten je poté ze všech buněk senzoru sekvenčně vyčten a převeden na napětí. Senzor má větší podíl světlocitlivé plochy, protože u jednotlivých pixelů se na křemíkové struktuře nenachází žádný převodník navíc. Převod na napětí, zesílení a převod na digitální signál se děje mimo světlocitlivou plochu čipu, takže ji nezmenšuje.
- **CMOS** – světlocitlivá buňka převede dopadající foton na elektrický náboj, elektrický náboj je hned v místě převeden na napětí, poté je signál vyčten najednou ze všech buněk a dále zpracován (zesílen, převeden na digitální). Podíl světlocitlivé plochy je menší, protože konverze náboje se děje přímo na čipu, převodník zabírá světlocitlivou plochu. [4], [5]



Obr. 3: Schematické porovnání CCD a CMOS  
Převzato z [5]

Dle uspořádání pixelů na senzoru rozlišujeme kamery:

- **řádkové**, které mají pouze jeden řádek pixelů, pro zachycení snímku objektu je nutné, aby se kamera nebo objekt pohyboval
- **plošné**, kde jsou pixely uspořádány v plošné matici

Senzory kamer mohou mít různý poměr stran a různé rozměry. Jejich velikost je charakterizována úhlopříčkou v palcích (typicky 1/3" - 1,1"). Senzory mají taky různý počet bodů, tj. různé rozlišení (typicky 0,3 Mpx – 20 Mpx) a tím pádem i různou velikost pixelu (v řádech mikrometrů).

### Citlivost senzoru na vlnové délky

Dle citlivosti senzoru kamery na jednotlivá pásma záření se kamery dělí na:

- **UV kamery** – citlivé na vlnové délky kratší, než viditelné záření ( $< 380$  nm)
- **Běžné kamery** – citlivé na viditelné záření
- **IR kamery** – citlivé na vlnové délky delší, než viditelné záření ( $740$  nm), dle konkrétního pásma se dále dělí na NIR (Near Infrared, blízké IR záření), SWIR (Short-Wavelength Infrared, krátkovlnné IR), MWIR (Mid-Wavelength Infrared, středněvlnné IR), LWIR (Long-Wavelength Infrared, dlouhovlnné IR).

Dle citlivosti jednotlivých pixelů na vlnové délky se kamery dělí na:

- **Monochromatické** – také černobílé, všechny body senzoru jsou citlivé na shodné spektrum vlnových délek. Ve strojovém vidění se používají nejčastěji, protože mají větší citlivost (díky absenci filtrů na pixelech senzoru) a malý datový tok (většinou 8 bitů/pixel).
- **Barevné** – pixely kamery jsou překryty barevnými filtry, tzv. Bayerovou maskou, ve které se střídají v určitém obrazci postupně červený, zelený a modrý filtr. Pro získání finálního barevného obrazu se obraz musí tzv. debayerizovat, tj. dopočíst pro každý pixel hodnoty

zbývajících dvou barev, což lze provést přímo v kameře (na FPGA), nebo při zpracování obrazu v nadřazeném systému (např. PC, za cenu vyššího využití CPU). Nevýhodou je vyšší datový tok v případě debayerizace v kameře (zpravidla 24 bitů/pixel) a menší citlivost senzoru daná přítomností filtrů na senzoru.

- **Multispektrální a hyperspektrální** – matice senzoru je rozdělena na menší submatice o až několika desítkách prvků. V každé submatici je každý pixel opatřen odlišným filtrem, který propouští velmi úzké pásmo vlnových délek. Z výsledného obrazu tak lze získat několik podobrazů (dle počtu prvků v submatici), kde každý z nich reprezentuje snímanou scénu v určitém malém pásmu vlnových délek. V dnešní době tato technologie nabývá na popularitě např. v souvislosti s moderním přístupem k zemědělství, kde pomáhá s určováním podílu vody v krajině. Příklad jedné submatice s vyznačenými vlnovými délkami v nm, které jednotlivé filtry propouští, je na obr. 4

Barevný či multispektrální obraz může být vytvořen také kamerou s optickým hranolem, který rozkládá světlo na jednotlivé barevné složky a láme je do různých směrů, a s více čipy umístěnými kolem hranolu. Taková kamera může mít např. 4 CCD snímače, které jsou citlivé na R, G, B a např. NIR složku světla. Výhodou pak je absence debayerizace a s ní souvisejících barevných vad na kontrastních přechodech, nevýhodou jsou větší rozměry kamery.

468	480	495	503
512	526	537	550
555	570	582	591
602	613	625	637

Obr. 4: Příklad masky na senzoru hyperspektrální kamery

## 2.2 Objektivy

Objektiv je soustava čoček, která umožňuje směřovat světlo z požadovaného směru na čip kamery, zajišťuje tak dostatečnou ostrost obrazu. V úpravě pro průmysl se jedná o robustně konstruovaný objektiv se zpravidla pevnou ohniskovou vzdáleností, manuálním ostřením a manuální clonou. Příklad objektivu pro strojové vidění je na obr. 5.



Obr. 5: Běžný megapixelový objektiv s ohniskovou vzdáleností 16 mm

### Ohnisková vzdálenost objektivu

Ohnisková vzdálenost je vzdálenost v milimetrech mezi středem optické soustavy objektivu a rovinou senzoru kamery. Určuje zorný úhel – platí, že čím delší ohnisková vzdálenost, tím menší zorný úhel. Dle schopnosti objektivu měnit ohniskovou vzdálenost rozlišujeme objektivy s pevnou ohniskovou vzdáleností a proměnnou ohniskovou vzdáleností (tzv. zoom objektivy), přičemž změna ohniskové vzdálenosti se může dít buď manuálně nebo motorizovaně.

### Světelnost, clona objektivu

Světelnost je schopnost objektivu propouštět světlo. Jedná se o bezrozměrné číslo, které vyjadřuje poměr ohniskové vzdálenosti k průměru vstupní čočky, zjednodušeně řečeno vyjadřuje, kolik světla projde objektivem až k senzoru kamery.

Clona je mechanické zařízení zpravidla v podobě několika lamel regulujících velikostí otvoru, který lamely tvoří, průchod světla objektivem. V průmyslovém objektivu se zvláštním důrazem na odolnost proti vibracím může být clona i pevná, neměnná.

Clonové číslo popisuje úroveň zaclonění objektivu. Minimální clonové číslo na objektivu je jeho světelnost. Zapisujeme jej ve tvaru např. F1.4, příp.  $f/1.4$ . Platí zde nepřímá kvadratická úměra, tj. 2x větší clonové číslo znamená 4x méně procházejícího světla. Také platí, že čím větší clonové číslo, tím větší hloubka ostroty. Příliš vysoké clonové číslo ovšem znamená velmi malý otvor ve cloně, což má za následek zhoršení ostroty kresby objektivu v důsledku difrakčních jevů. [1]

Každý objektiv má svůj tzv. sweet spot, tj. optimální clonové číslo, při kterém dosahuje nejlepší kresby. Při vyšším clonovém čísle je objektiv náchylný k difrakčním jevům a obraz je tak neostrý, naopak při plně otevřené cloně se projeví především chromatická aberace a vinětace (popis vad viz dále).

### Rozlišení objektivu

Rozlišení objektivu je možno udat více způsoby. Jedním způsobem je parametr lp/mm, který udává, kolik párů střídajících se černých a bílých linek je objektiv schopen vykreslit na čip.

Závislost kontrastu obrazu na počtu čar na milimetr se udává v tzv. MTF grafu. Rozlišuje se počet párů linek v tangenciálním a radiálním směru. Druhou možností je uvedení rozlišení v megapixelech. Objektivy takto mohou být označeny např. jako megapixelové, 5megapixelové atp. Aby tento parametr měl význam, nutno znát také velikost čipu, pro který je objektiv určen.

### Objektiv a velikost čipu

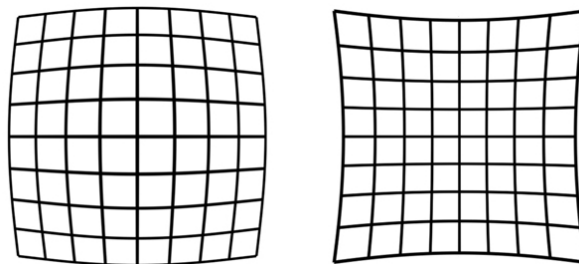
Každý objektiv má určenou maximální velikost senzoru kamery, pro který je určen. Je totiž schopen vytvořit obraz ve tvaru kruhu o konečném průměru. Bez problému lze použít objektiv určený pro větší senzor na kameře s menším senzorem. Pokud ovšem dojde k použití objektivu určeného pro menší snímač na kameře s větším snímačem, dojde na okrajích obrazu k vinětaci, případně k úplnému ořezu obrazu.

### Zorné pole objektivů

- Širokoúhlé objektivy – objektivy se širokým úhlem záběru (až  $180^\circ$ ), neexistuje univerzální kritérium, podle kterého se objektiv do této kategorie řadí.
- Teleobjektivy – objektivy s velmi malým zorným úhlem.
- Makroobjektivy – objektivy pro snímání malých předmětů z malých vzdáleností, mají velké zvětšení.
- Telecentrické objektivy – speciální objektivy, které mají schopnost potlačovat perspektivu, protože jimi prochází pouze paprsky rovnoběžné s optickou osou. Využívají se pro přesné měření rozměrů (nejlépe v kombinaci s koaxiálním či kolimovaným osvětlením).
- Pericentrické objektivy – speciální objektivy, které umožňují z jednoho místa zobrazit jak horní část předmětu, tak jeho boční plochy.

### Vady objektivů

- **Chromatická aberace** je barevná vada objektivu, způsobená závislostí indexu lomu na vlnové délce světla. V praxi se nejvíce projevuje v okrajových částech obrazu jako barevný přechod na černobílém kontrastním přechodu.
- **Zkreslení** je geometrická vada objektivu, kdy rovné linie jsou objektivem reprezentovány jako křivky. Dle tvaru křivek rozlišujeme poduškové a soudkové zkreslení, viz obr. 6.
- **Vinětace** je vada, při které ubývá intenzita světla směrem k okrajům obrazu.



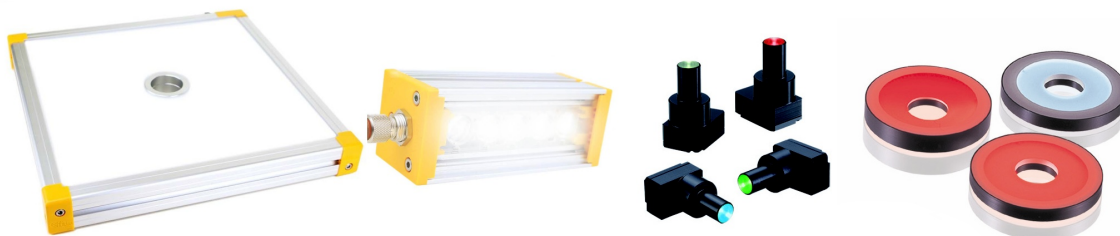
Obr. 6: Příklad soudkového a poduškového zkreslení objektivu

### Pracovní vzdálenost

Pracovní vzdálenost je vzdálenost mezi přední čočkou objektivu a snímaným objektem, resp. rovinou zaostření. Důležitá je minimální pracovní vzdálenost, na kratší vzdálenost již objektiv není schopen bez dodatečných součástí (viz dále) zaostřit. Objektivy bez možnosti ostření mají pracovní vzdálenost pevně danou (např. většina makroobjektivů a telecentrických objektivů).

### 2.3 Osvětlovače

Neméně důležitou součástí sestavy pro strojové vidění jsou osvětlovače, správně zvolený osvětlovač totiž může podstatným způsobem ulehčit následné zpracování obrazu a to tak, že hledané vady/parametry charakterem osvětlení zvýrazní, a naopak ostatní vlastnosti objektu potlačí. Při volbě osvětlovače je nutno zvážit zejména tvar, velikost, umístění, povrch a barvu snímaného objektu. Příklady osvětlovačů jsou na obr. 7, zleva plošný osvětlovač s otvorem pro kameru, liniový osvětlovač, bodové a kruhové osvětlovače.



Obr. 7: Příklad různých typů osvětlovačů

### Spektrum osvětlovače

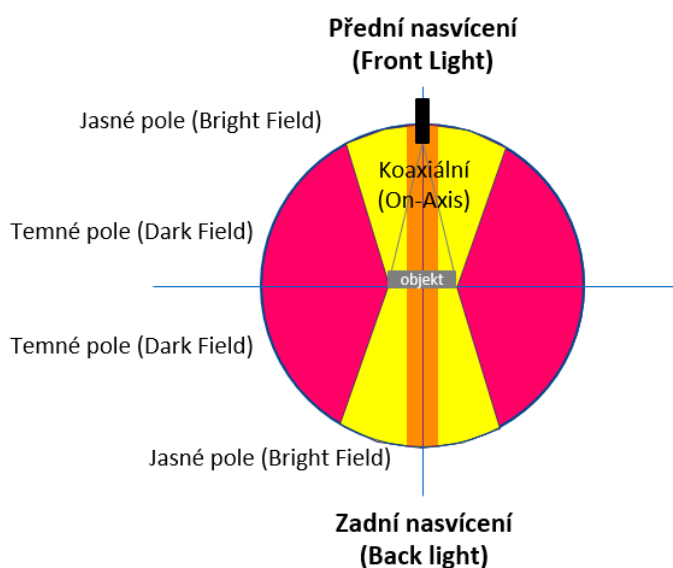
Osvětlení může být buď monochromatické, tj. vyzařující na jedné světelné délce, nebo bílé/šírokospektrální. Osvětlovač může vyzařovat i mimo oblast viditelného záření. Spektrum záření světla je vždy nutno volit vzhledem k užití kamery (z důvodu její spektrální citlivosti) a ke snímanému objektu.



## Směr a pozice osvětlení

Dle umístění osvětlovače rozlišujeme několik typů (viz obr. 8):

- **Čelní osvětlení** – osvětlovač svítí na objekt ze stejné strany, jako je umístěna kamera.
- **Zadní osvětlení** – osvětlovač je umístěn ze zadní části zkoumaného objektu, proti kameře. Objekt se tak jeví jako tmavý na bílém pozadí. Vhodné pro inspekci rozměrů a kontur objektu.
- **Koaxiální osvětlení** – osvětluje objekt paprsky rovnoběžnými s optickou osou objektivu. Velmi vhodné při použití s telecentrickým objektivem. Používá se např. při inspekci povrchů, popisů, DPS atp.
- **Jasně pole** – osvětluje objekt převážně kolmo na rovinu snímání. Používá se v širokém rozsahu úloh.
- **Temné pole** – osvětluje objekt ve směru roviny snímání. Využívá se zejména pro inspekci povrchu, reliéfů, protože zvýrazňuje prostorové vady povrchu, je inverzní ke koaxiálnímu.
- **Dómové osvětlení** – osvětlovač ve tvaru polokoule s malým otvorem pro kameru a velkým otvorem pro kontrolovaný objekt zajišťuje velice homogenní, měkké světlo, čehož se využívá při inspekci lesklých materiálů pro minimalizaci odlesků.



Obr. 8: Schematický popis možných umístění osvětlovače

Dle směrovosti osvětlení rozdělujeme světlo:

- **Směrové** – osvětlovač emituje světlo velmi intenzivně pouze v jednom směru.
- **Difúzní** – osvětlovač emituje světlo homogenně, tj. ve více směrech, nepřímě.

Charakter osvětlení je do jisté míry dán již samotnou konstrukcí, ale lze jej dále ovlivňovat např. umístěním dodatečného difuzoru, změnou pozice čoček (umožňuje-li to konstrukce osvětlovače) atp.

## Zdroje světla

- **LED** – v současné době naprosto převažující technologie osvětlení, vhodná pro většinu aplikací. Výhodou je kompaktnost jedné LED, proto je možné je sestavovat do různých geometrických uspořádání. Další výhodou jejich spolehlivost a poměrně dlouhá životnost.
- **Zářivky** – zřídka používané, je nutno potlačit stroboskopický efekt dostatečně vysokou frekvencí napájení.
- **Žárovky** – dnes se pro tuto oblast již prakticky nepoužívají.
- **Výbojky** – jejich výhodou je široké spektrum vyzařovaných vlnových délek.
- **Lasery** – využívají se zejména pro projekci obrazců (např. linie pro 3D laserovou triangulaci, mřížky a matice pro detekci deformací, vzdáleností atp.).

Při regulaci jasu zdrojů je nutno zajistit, aby nedocházelo ke změně jasu obrazu v čase (blikání obrazu), proto se používá spojitá regulace jasu, ne PWM.

## Konstrukce osvětlovačů

- **Plošné** – světlo je vyzařováno z plochy (a to buď nepřímě anebo např. z LED sestavených do matice).
- **Liniové** – světlo je vyzařováno např. z jedné nebo více řady LED, které jsou uspořádány tak, aby vytvářely podlouhlou světelnou stopu.
- **Bodové** – světlo je soustředěno za pomoci optických členů do jednoho místa.
- **Kruhové** – světlo je vyzařováno z osvětlovače ve tvaru mezikruží, v jehož středu je zpravidla umístěna kamera.

## Spojítost vyzařování

- **Spojité** – osvětlovač vydává světlo nepřetržitě po celou dobu, kdy je na něm připojeno napájení. Výhodou je absence kabeláže mezi nadřazeným řídicím systémem či kamerou, ovšem zpravidla za cenu snížení životnosti osvětlovače.
- **Stroboskopické** – osvětlení je triggerováno pouze po dobu expozice, spouštění může řídit nadřazený systém nebo samotná kamera za pomoci svého výstupu a kontroléru pro konkrétní výrobní řadu osvětlovačů.

## 2.4 Doplnující součásti sestavy pro strojové vidění

Sestava pro strojové vidění může mít mnoho dalších částí, které jsou užity v případě, že je aplikace vyžaduje.

- **Filtry** – umožňují potlačit vliv některých vlnových délek (barevné filtry, IR, UV filtry) nebo propustit pouze polarizované světlo (lineárně či cirkulárně polarizační filtry).

- Mezikroužky – umožňují objektivu zaostřit na kratší vzdálenost za cenu snížení hloubky ostrosti a nemožnosti zaostřit na nekonečno.
- Telekonvertory – umožňují prodloužit ohniskovou délku objektivu o násobek (většinou v rozsahu 1,4 – 4x) za cenu ztráty světelnosti objektivu.

## 2.5 Expozice

Úkolem sestavy pro strojové vidění je sejmout adekvátně jasný obraz v dostatečné kvalitě. Část obrazu, která nese informaci o snímaném objektu nesmí být podexponovaná (hodnota jasu 0) nebo přexponovaná (hodnota jasu 255). Sejmutí obrazu se nazývá též expozice a lze ji ovlivnit několika parametry:

- Clona – čím větší clonové číslo, tím méně světla dopadá na senzor a tím větší hloubka ostrosti.
- Čas – čím delší expoziční čas, tím více světla dopadá na senzor a tím větší pohybová neostrost.
- Citlivost – čím vyšší citlivost, tím jasnější obraz, ale vyšší šum a menší dynamický rozsah.

Úkolem experta je mj. výše uvedené složky správně vybalancovat, aby obraz splňoval požadavky na něj kladené.

### 3 Popis vývojových prostředí/technologií používaných pro návrh kamerových systémů – přehled a charakteristika

Pro zpracování obrazu je možno využít hned několika vývojových prostředí (technologií, programovacích jazyků), která mají k dispozici knihovny pro strojové vidění a další nástroje pro ulehčení práce s obrazem. Každé z nich má své výhody s ohledem na zamýšlenou oblast použití. Na následujících řádcích představím některá z nich, nejedná se ovšem o vyčerpávající přehled.

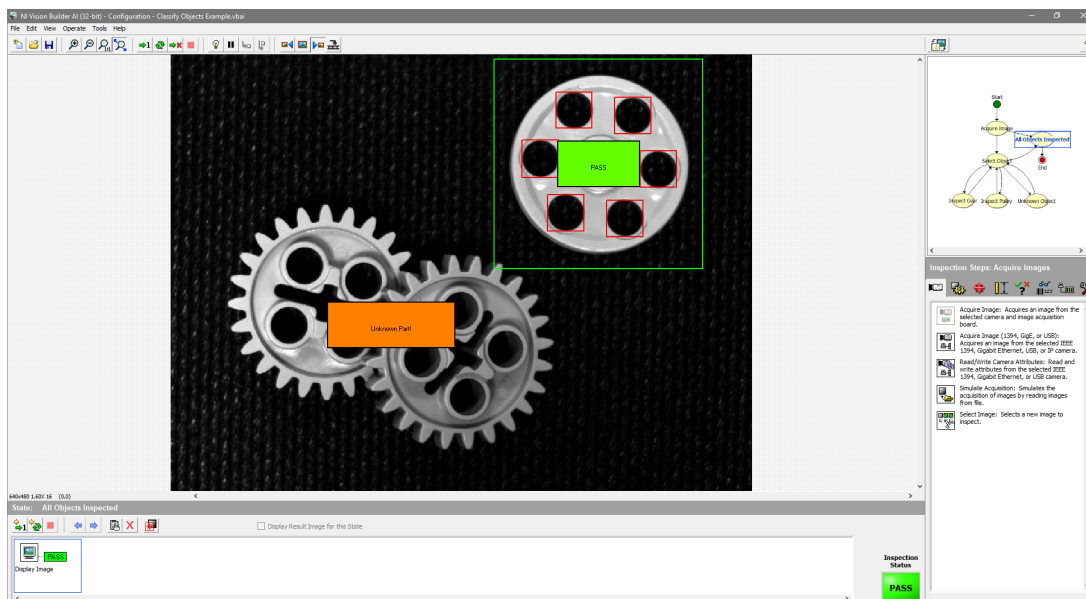
#### 3.1 Vision Builder for Automated Inspection

Vision Builder for Automated Inspection (dále VBAI) je SW nástroj od společnosti National Instruments, který slouží pro vytváření a nasazení tzv. inspekce strojového vidění (viz obr. 9). Inspekce je posloupnost kroků, které umožňují např.:

- Sejmout snímek z kamery nebo jej načíst z disku.
- Upravit snímek (prahování, úprava kontrastu, filtrování atp.).
- Kalibrovat snímek pro provádění měření v reálných jednotkách.
- Provádět zpracování obrazu pro strojové vidění, např. detekce hran, objektů, hledání vzorů, sledování objektů, měření intenzity v obrazu, měření rozměrů, porovnávání křivek, čtení znaků, 1D a 2D kódy (barcode, QR), klasifikace objektů.
- Komunikovat s okolními zařízeními přes analogové a digitální vstupy, sériovou linku, Ethernet.
- Ukládat snímky a výsledky jednotlivých částí inspekce.
- Volat kód z LabVIEW.
- Využívat větvení programu do uživatelem definovaných stavů a rozhodovací logiku na základě aktuálních výsledků inspekce.

Z pomocných nástrojů je k dispozici například automatické generování kódu LabVIEW, výsledný kód je ovšem značně neefektivní, a tak se hodí spíše jako startovací bod k optimalizaci. Oproti LabVIEW nejsou ve VBAI pro zpracování obrazu dostupné některé komplikovanější funkce. Inspekci je možno podrobit výkonnostnímu testu pro zjištění maximální frekvence zpracování obrazu na daném PC. Tvorba inspekce probíhá v tzv. konfiguračním módu, po dokončení lze aplikaci přepnout do inspekčního módu, ke kterému přísluší i GUI, které je taktéž možno upravovat.

Celé prostředí je navrženo tak, aby jej mohl používat i uživatel bez programátorských zkušeností, což se vyznačuje zejména jednoduchostí veškerého nastavení a tvorby inspekce. Velmi výhodné je využití VBAI zejména pro jednoduché aplikace nebo pro ověření navržené koncepce zpracování obrazu. Pro pokročilejší aplikace je, vzhledem např. k absenci komplikovanějších funkcí nebo nemožnosti paralelismu, nevhodný.

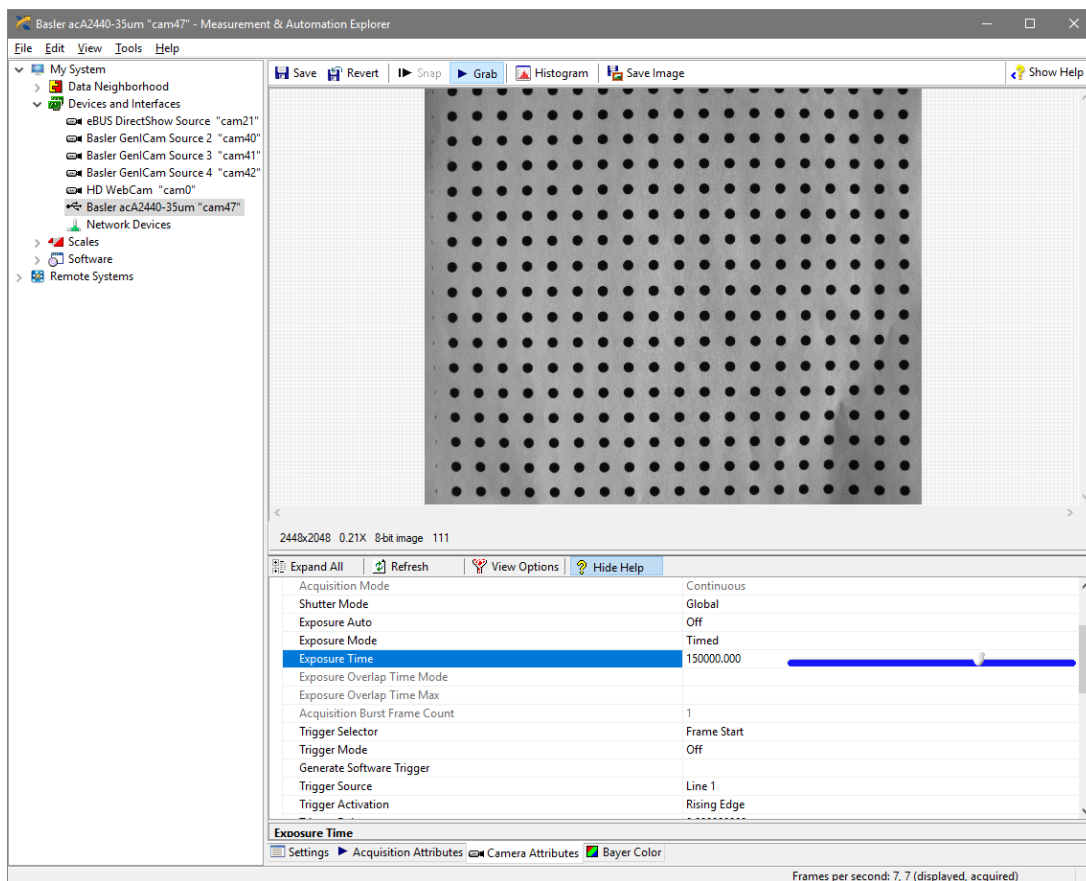


Obr. 9: Konfigurační rozhraní VBAI

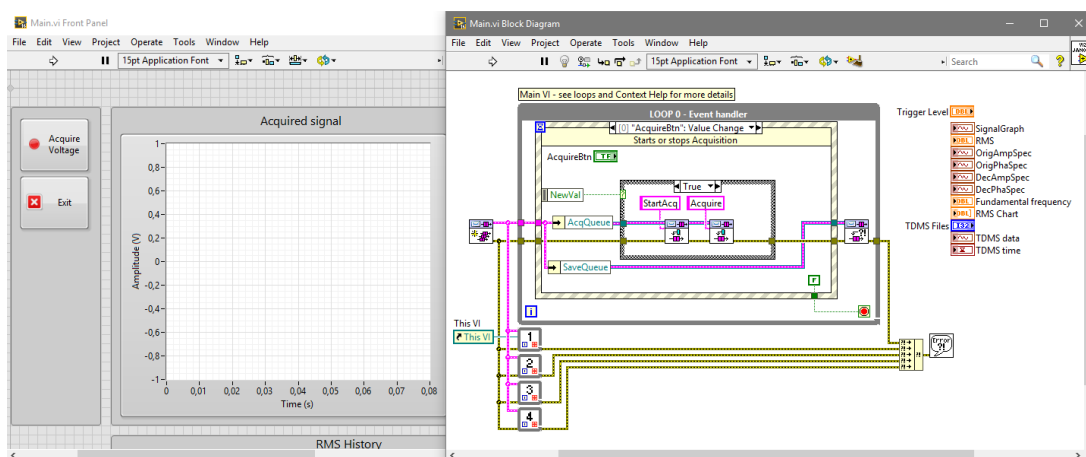
Rozhraní dominuje zpracovávaný snímek, v dolním panelu jsou kroky inspekce. Vpravo nahoře se nachází stavový diagram inspekce, dole paleta možných funkcí pro úpravu obrazu.

### 3.2 LabVIEW

LabVIEW je grafický programovací jazyk, který vznikl v roce 1986 v americké firmě National Instruments. Stejnomené je i vývojové prostředí pro tento programovací jazyk. Tento programovací jazyk byl vytvořen pro inženýry z oblasti měření/řízení/automatizace, kteří nemají zkušenosti s programováním. Silná stránka tohoto jazyka je mj. velmi dobrá SW podpora v podobě množství knihoven určených pro komunikaci s různým typem hardware, měřicích přístrojů, dále pro zpracování signálů atp. Mezi tyto knihovny se řadí i Vision Acquisition Software (knihovna IMAQdx, IMAQ), což je ovladač pro kamery, a Vision Development Module, který slouží pro úpravy a zpracování obrazu. V porovnání s VBAI v oblasti strojového vidění nabízí komplexnější a pokročilejší funkce. Pro konfiguraci HW, tedy i kamer, je určený konfigurační nástroj s názvem Measurement & Automation Explorer, který je součástí softwarového balíku od NI. Ten umožňuje snímat obraz a nastavovat veškeré parametry kamery. [6], [7], [8], [9]



Obr. 10: Measurement & Automation Explorer – parametry kamery

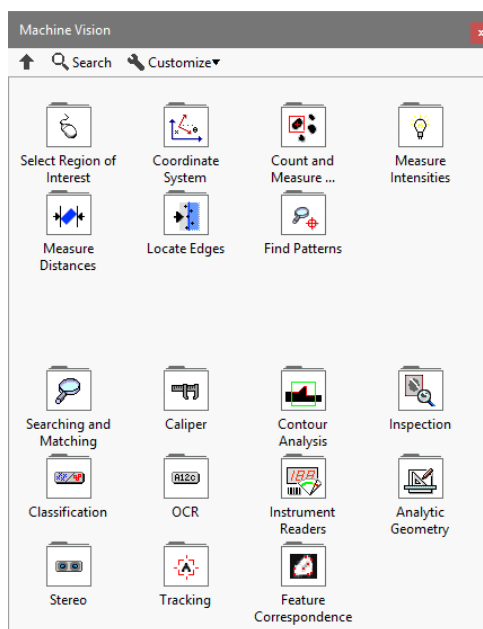


Obr. 11: Příklad čelního panelu a blokového diagramu

Příklady některých palet funkcí, které nabízí modul VDM pro LabVIEW (viz obr. 12), jsou:

- „**Processing**“ – obsahuje VI pro prahování, úpravu jasu, kontrastu obrazu, korekce nerovnoměrnosti osvětlení.

- „**Filters**“ – obsahuje funkce pro filtraci a konvoluci.
- „**Morphology**“ – umožňuje detekovat objekty v obraze a provádět s nimi různé operace (zvětšovat, zmenšovat, filtrovat je atd.).
- „**Analysis**“ – funkce této palety slouží pro statistickou analýzu obrazu (histogram, profily křivek, klasifikace objektů atd.).
- „**Color Processing**“ – funkce pro práci s barevným obrazem (histogram, prahování, klasifikace barev atd.).
- „**Operators**“ – umožňuje dělat matematické a logické operace s obrazy (sčítání, odečítání, násobení, logické součet, součin atd.).
- „**Frequency Domain**“ – funkce pro zpracování obrazu ve frekvenční doméně, např. FFT, inverzní FFT, operace se snímky v oboru komplexních čísel.
- „**Measure Distances**“ – funkce pro měření vzdáleností mezi objekty a rozměry objektů.
- „**Locate Edges**“ – funkce pro hledání hran.
- „**Searching and Matching**“ – funkce pro učení a hledání vzorů v obraze (na základě geometrických vlastností, barevnosti či kontury objektu).
- „**OCR**“ – funkce pro čtení znaků v obraze.
- „**Instrument Readers**“ – paleta specializovaná na čtení digitálních či analogových ukazatelů měřicích přístrojů a na čtení 1D a 2D kódů.



Obr. 12: Paleta funkcí „Machine Vision“ z modulu VDM v LabVIEW

### 3.3 OpenCV

OpenCV, z anglického Open Source Computer Vision Library, tj. otevřená knihovna pro strojové (počítačové) vidění, je zdarma dostupná knihovna psaná v C/C++ pro zpracování obrazu, s důrazem na real-time a vícevláknové vykonávání. Obsahuje několik stovek algoritmů pro strojové vidění, od tradičních (detekce hran, prahování, filtrace atp.) až po nejmodernější – např. strojové učení. Využití těchto algoritmů je velmi široké – pro identifikaci objektů, tváří, sledování objektů, tvorba 3D obrazů, spojování obrazů atp. Disponuje rozhraními pro C++, Javu a Python, podporuje vícero platforem (Windows, Linux, Mac OS, iOS, Android). Vzhledem ke své komplexnosti a univerzálnosti má obří uživatelskou základnu po celém světě. [10]



## 4 Rozbor funkce a popis algoritmů strojového vidění pro požadované optické testy SMD součástky

Strojové vidění využívá pro zpracování obrazu nespočet různých algoritmů, vzhledem k rychlému vývoji zejména v poslední době se navíc jejich množství stále zvyšuje. V této kapitole budou popsány pouze některé základní, zejména s ohledem na praktickou část diplomové práce.

### 4.1 Segmentace

Segmentace obrazu je proces, při kterém se obraz dělí do částí, které mají korelaci s objekty nebo plochami reálného světa zachycenými na snímku. Rozlišujeme kompletní segmentaci, jejímž výsledkem je množina oddělených oblastí v obraze, kde každá z oblastí reprezentuje některý z objektů v obraze, nebo částečnou segmentaci, kde oblasti přesně nerepresentují objekty v obraze. Kompletní segmentace je popsána:

$$R = \bigcup_{i=1}^S R_i, \quad R_i \cap R_j = \emptyset, \quad i \neq j, \quad (1)$$

kde  $R$  je obraz,  $R_1 \dots R_S$  je konečný počet oblastí v obraze. Segmentaci obrazu je možno docílit různými méně či více náročnými technikami zpracování obrazu, např. prahováním, hledáním vzorů, segmentací na základě hran atp. [11]

### 4.2 Prahování

Pro analýzu obrazu se velmi často používá binární reprezentace snímku. Pro převod monochromatického snímku na binární se používá tzv. prahování. Matematicky lze globální prahování popsat jako transformaci vstupního snímku  $f$  na výstupní snímek  $g$ :

$$\begin{aligned} g(i, j) &= 1 \text{ pro } f(i, j) > T, \\ &= 0 \text{ pro } f(i, j) \leq T, \end{aligned} \quad (2)$$

kde  $T$  je práh (threshold),  $i$  a  $j$  jsou indexy v matici obrazu,  $g(i, j) = 1$  pro objekty a  $g(i, j) = 0$  pro pozadí (nebo naopak).

Nejdůležitějším faktorem pro docílení správné segmentace obrazu za pomoci prahování je správný výběr prahu. Pouze výjimečně lze docílit kompletní segmentaci obrazu globálním prahováním. V jiných případech je nutno přistoupit k adaptivnímu lokálnímu prahování. Zatímco globální prahování celého snímku  $f$  je popsáno:

$$T = T(f), \quad (3)$$

z čehož vyplývá, že globální práh je funkcí pouze celého snímku, zatímco lokální prahování je závislé na pozici v obraze:

$$T = T(f, f_c), \quad (4)$$

kde  $f_c$  je část obrazu, ze které je určen práh. Základní rovnice 2 má několik různých modifikací. Jednou z nich je modifikace pro segmentaci oblastí z množiny šedých obrazových bodů  $D$ : [6], [11]

$$\begin{aligned} g(i, j) &= 1 \text{ pro } f(i, j) \in D, \\ &= 0 \text{ pro ostatní } f(i, j). \end{aligned} \quad (5)$$

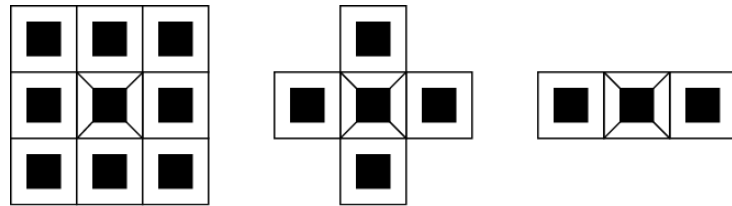
### 4.3 Matematická morfologie

Morfologické operace zjednodušují snímky, potlačují nechtěné částice a artefakty ve snímku při současném zachování důležitých charakteristik hlavních objektů. Morfologické funkce se využívají za různými účely, např.:

- předzpracování obrazu (filtrace šumu, zjednodušení tvarů),
- zvýraznění struktury objektu (skeleton, ztenčení atp.),
- segmentace obrazu.

Morfologie užívá metod nelineární algebry, oproti tradičnímu nemorfologickému přístupu (lineární transformace, konvoluce) je ve většině případů rychlejší a pro některé aplikace podává lepší výsledky. [11]

Morfologická transformace  $\Psi$  je dána vztahem snímku (reprezentovaném množinou bodů  $X$ ) a malou množinou bodů  $B$ , která se nazývá strukturní element.  $B$  je vztažena k lokálnímu počátku souřadnic  $O$  (reprezentační bod). Aplikace morfologické transformace  $\Psi(X)$  na obraz  $X$  pak znamená postupný posun strukturního elementu  $B$  po celém obraze (viz obr. 13). Pokud je  $B$  umístěno v obraze, pixel obrazu, který se nachází pod bodem  $O$  se nazývá současný pixel. Výsledek operace mezi  $X$  a  $B$  v současné pozici (který může být v binárním obraze pouze „0“ nebo „1“), je uložen ve výstupním obraze v současném pixelu.



Obr. 13: Příklad strukturních elementů

Morfologické operace jsou duální, protože ke každé množině existuje její doplněk, pro každou morfologickou operaci  $\Psi(X)$  pak existuje duální transformace  $\Psi^*(X)$ :

$$\Psi(X) = \left( \Psi^*(X^C) \right)^C \quad (6)$$

### 4.3.1 Dilatace

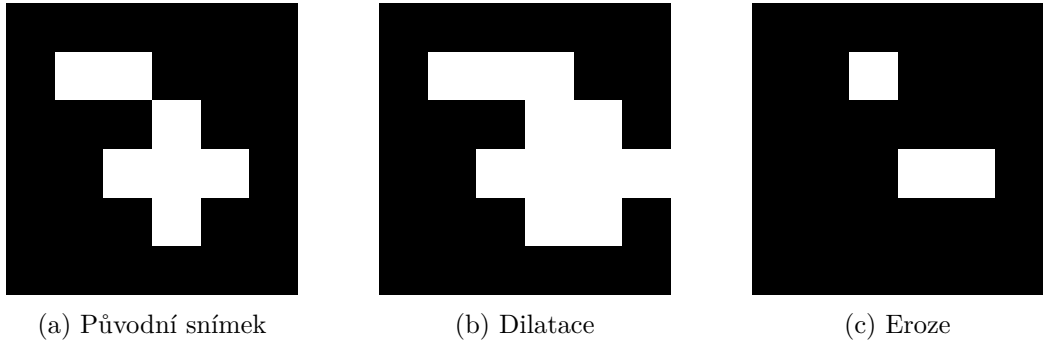
Dilatace se označuje symbolem  $\oplus$ . Slučuje dvě množiny za pomoci vektorového součtu. Dilatace množinou všech možných vektorových součtů párů prvků, kde jeden prvek je z  $X$  a druhý z  $B$ :

$$X \oplus B = \{p \in \mathcal{E}^2 : p = x + b, x \in X \text{ a } b \in B\} \quad (7)$$

Dilatace je komutativní, asociativní, invariantní vůči posunu a rostoucí:

$$\begin{aligned} X \oplus B &= B \oplus X, \\ X \oplus (B \oplus D) &= (X \oplus B) \oplus D, \\ X_h \oplus B &= (X \oplus B)_h, \\ X \subset Y, \text{ pak platí } X \oplus B &\subset Y \oplus B. \end{aligned} \quad (8)$$

Výsledek operace se liší dle strukturního elementu, obecně se jedná o vyplnění děr a zálivů a také zvětšení objektu. [6]



Obr. 14: Příklad morfologických operací  
Strukturní element  $B = \{(0, 0), (1, 0)\}$

### 4.3.2 Eroze

Eroze se označuje symbolem  $\ominus$ . Slučuje dvě množiny za pomoci vektorového rozdílu, je duální operací k dilataci.

$$X \ominus B = \{p \in \mathcal{E}^2 : p + x \in X \text{ pro každé } b \in B\} \quad (9)$$

Eroze je také invariantní vůči posunu, pokud je reprezentativní bod součástí strukturního elementu, pak je antiextenzivní a je také rostoucí:

$$\begin{aligned}
X_h \ominus B &= (X \ominus B)_h, \\
(0, 0) \in B, \text{ pak } X \ominus B &\subset X, \\
X \subset Y, \text{ pak platí } X \ominus B &\subset Y \ominus B.
\end{aligned} \tag{10}$$

Výsledek eroze se liší dle strukturního elementu, obecně se jedná o zjednodušení struktury objektu a vymazání úzkých částí objektu [11].

#### 4.4 Detekce hran

Detekce hran se využívá pro nalezení oblastí snímku, kde dochází k velkým změnám v intenzitě sousedních bodů. Na základě těchto změn lze nalézt kontury objektu a poté měřit jeho rozměry. Hrana je vlastnost přisouzená jednomu bodu v obraze, vypočítá se z obrazové funkce v jeho okolí. Jedná se vlastně o vektorovou veličinu, se svou velikostí a směrem. [6], [11]

##### 4.4.1 Algoritmus detekce hran aproximující derivaci obrazu za pomoci difference

Tyto algoritmy jsou většinou založeny na konvoluci, využívají různé konvoluční masky (operátory) o různé velikosti a různé citlivosti na směr hran. [11]

- **Robertsův operátor** – Jeden z nejstarších, nejjednodušších na výpočet, vzhledem ke své velikosti 2x2. Nevýhodou je velká citlivost na šum, protože se pro aproximaci gradientu používá malé množství okolních pixelů.

$$h_1 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} h_2 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \tag{11}$$

- **Laplaceův operátor** – Aproximuje druhou derivaci, určuje pouze sílu hrany, ne její směr. Nejčastěji se používá ve formě masky 3x3:

$$h = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \tag{12}$$

Výhodou je necitlivost vůči rotaci, nevýhodou je dvojitě zvýraznění některých hran. Někdy se používá ve formě masky s upravenou váhou prostředního pixelu a jeho okolí, pak ale ztrácí svou necitlivost vůči rotaci.

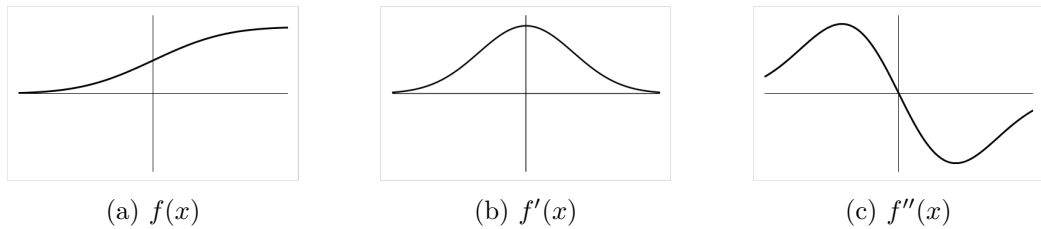
$$h = \begin{bmatrix} -1 & 2 & -1 \\ 2 & -4 & 2 \\ -1 & 2 & -1 \end{bmatrix} \tag{13}$$

- **Prewittův operátor** – Aproximuje první derivaci v osmi možných směrech (pro masku 3x3).

$$h_1 = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} h_2 = \begin{bmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix} \dots \quad (14)$$

#### 4.4.2 Algoritmus detekce hran založený na průchodu druhé derivace nulou

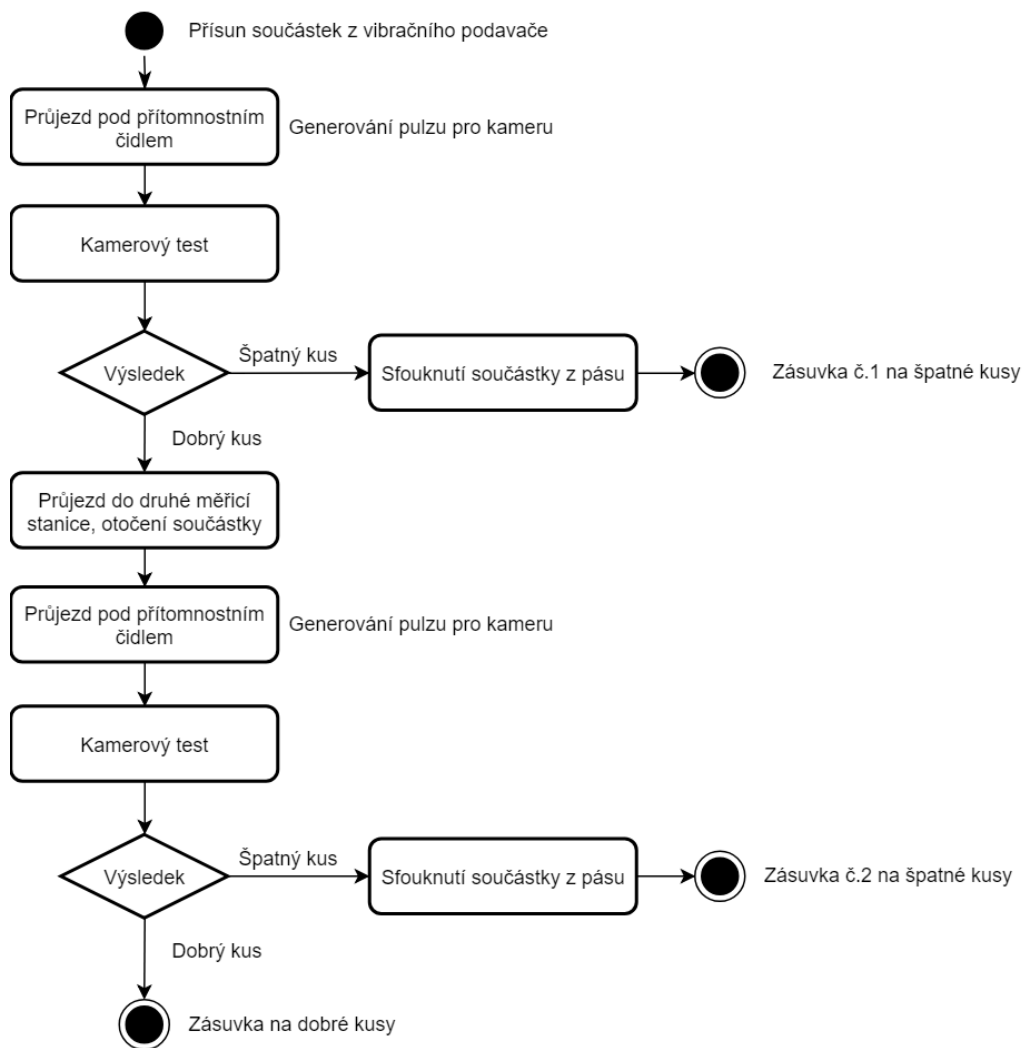
Tento algoritmus využívá faktu, že hrana v obraze znamená vysokou rychlost změny obrazové funkce, viz obr. 15. V první derivaci se hrana projeví lokálním extrémem v místě hrany, v druhé derivaci se tak projeví průchodem nulou. Protože hledání extrémů přímo z funkce je algoritmicky náročné, využívá se pro hledání hran právě hledání průchodu nulou druhé derivace. Před výpočtem druhé derivace se obraz filtruje, aby došlo k vyhlazení lokálních hran (např. vlivem šumu v obraze). [11]



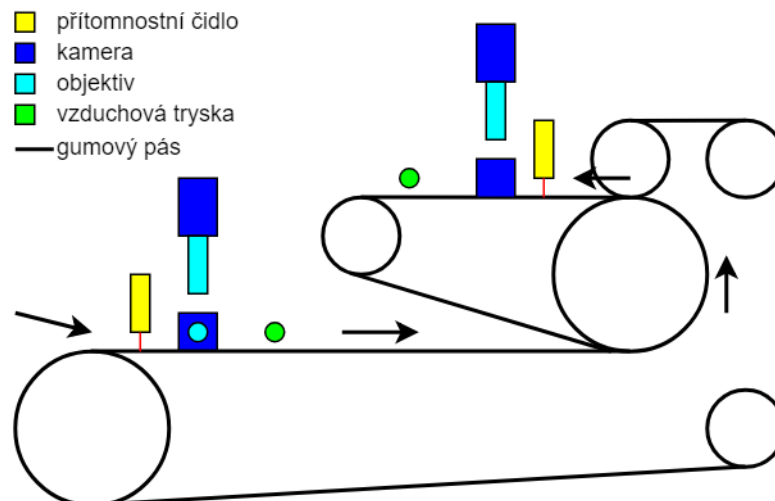
Obr. 15: Reprezentace hran v 1D obrazové funkci

## 5 Popis testeru

Tester slouží pro optickou inspekci SMD součástek. Tyto součástky se do stroje dostávají z **vi-bračního zásobníku**, který součástkami plní obsluha. Z něj se součástky dostávají do **kruhového vibračního podavače**, který má za úkol součástky naorientovat do správné pozice; v případě, že je více součástek na sobě, horní oddělit tak, aby zůstala pouze jedna; a seřadit součástky za sebe. Následuje **vibrační dopravník**, ze kterého se součástky dostávají na **gumový pás**.



Obr. 16: Diagram popisující průjezd součástky strojem.



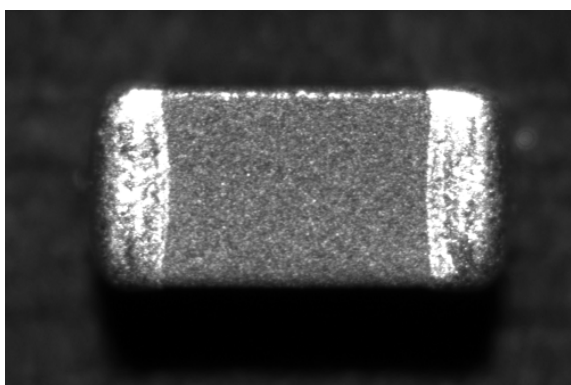
Obr. 17: Zjednodušené schéma stroje

Průjezd součástky je znázorněn na obr. 16. Na gumovém páse projíždějí součástky rychlostí cca 10 cm/s dvěma tzv. **kontrolními stanicemi**. V nich jsou nejprve součástky zarovnány **zarovnávacím kovovým kolečkem a vodítkem**, které mají za úkol zajistit stabilní pozici na páse v příčném směru. U zarovnávacího vodítka je **optické přítomnostní čidlo**, které detekuje průjezd součástky. Za nimi následuje **dvojice kamer s osvětlovači**, které snímají součástku ze dvou stran (jedné horní a jedné boční). Na konci kontrolní stanice je rychlý **vzduchový ventil s tryskou**, který sfoukává vadné součástky do krabíčky pro ně určené. Bezvadné součástky jsou za pomoci druhého gumového pásu otočeny tak, aby bylo možno zkontrolovat tu stranu součástky, na které součástka doposud ležela na páse. Po otočení součástky následuje průjezd druhou kontrolní stanicí, která funguje analogicky k první, pouze kontrola probíhá na zbylých dvou stranách součástky (dvě nejmenší čelní strany jsou bez kontroly). Na konci pásu se bezvadné součástky dostávají skluzem do krabíčky určené pro bezvadné součástky. Celý inspekční prostor stroje je kryt dveřmi s magnetickými čidly zavření. Zjednodušené schéma stroje je k dispozici na obr. 17.

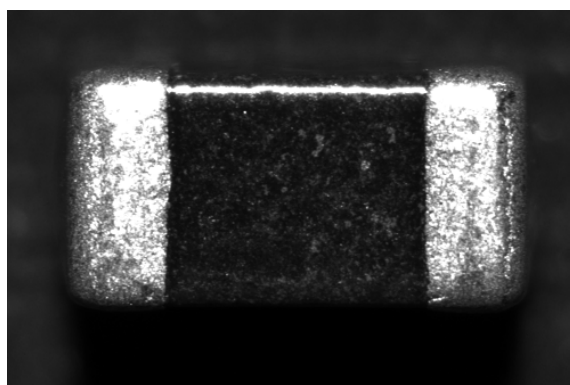
## 6 Teoretický rozbor a návrh řešení kamerové inspekce SMD součástky

### 6.1 Popis hlavních požadavků zákazníka

- *Inspekce součástek velikostí 0805 a 1206.* Tyto velikosti odpovídají standardním velikostem pouzder SMD, jsou udávány v setinách palců – velikost 0805 tak znamená rozměry 0,08 x 0,05 palců, tj. cca 2 x 1.25 mm (délka x šířka).
- *Inspekce součástek s keramickým a skleněným tělem.* Tělo součástky je ohraničeno dvěma terminacemi a může být buď keramické nebo potažené vrstvou skla. To způsobuje rozdíly při snímání obrazu, projevují se jinou intenzitou v obraze a odlesky – viz obr. 18a a 18b.
- *Kontrola součástek ze 4 stran (obě boční a obě horní).* Pro tyto kontroly budou využity 4 kamery, dvě boční a dvě horní. Dvě nejmenší čelní strany jsou bez kontroly.
- *Maximální takt 6–8 součástek za sekundu.*
- *Kontrola optických vad terminací, optických vad na těle součástky (ve skle anebo keramice) a rozměrů čipu.*



(a) Součástka s keramickým tělem

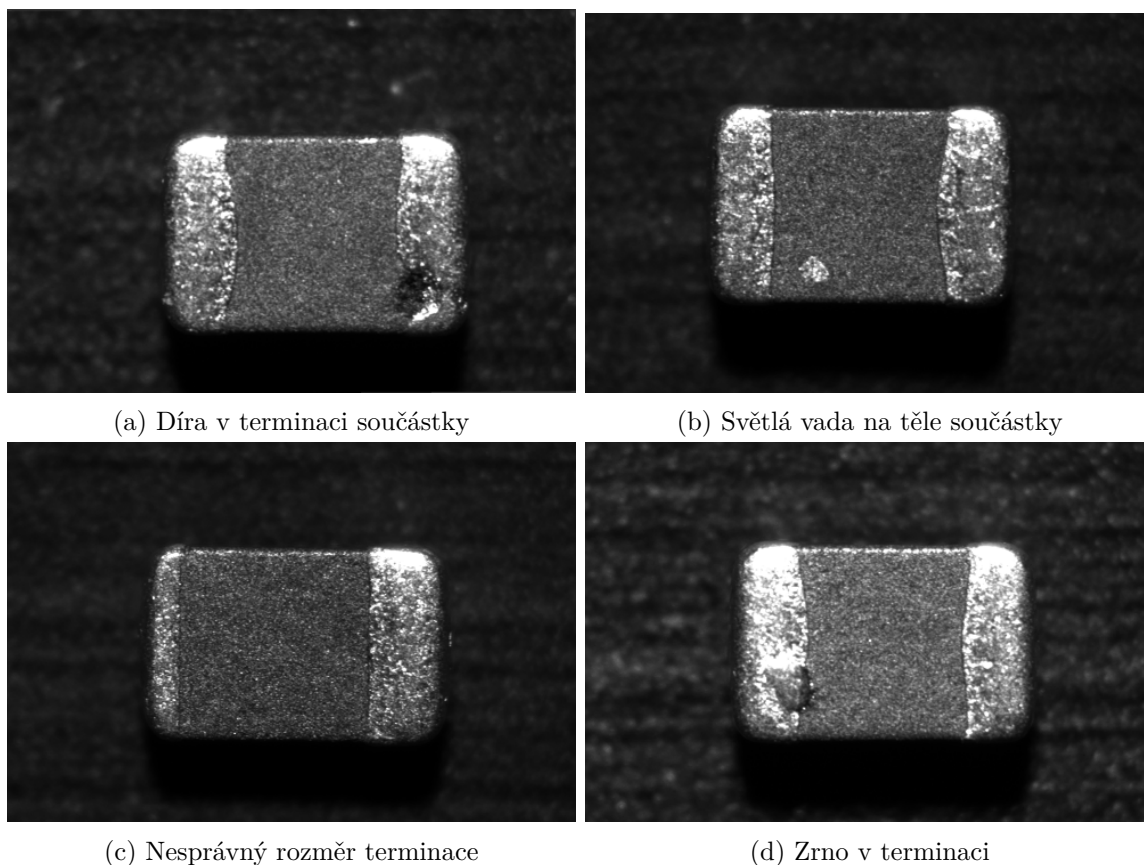


(b) Součástka se skleněnou vrstvou

Obr. 18: Snímky kontrolovaných součástek

Níže jsou uvedeny některé z vad, které mají být na součástkách kontrolovány. Snímky vad v plném rozlišení jsou součástí přílohy DP.





Obr. 19: Příklady kontrolovaných vad součástek

## 6.2 Teoretický rozbor kamerové inspekce

### 6.2.1 Hardware

#### Osvětlení

Osvětlení musí dostatečně silné, aby umožnilo dostatečné nasvícení projíždějící součástky i při velmi krátkém čase (řádově desítky  $\mu\text{s}$ ), musí také co nejvíce zvýraznit terminace součástky a hledané vady. Osvětlení, včetně případné optiky, musí být co nejkompaktnějších rozměrů, aby bylo možné jej do poměrně stísněného prostoru stroje umístit. Taktéž musí umožňovat (samostatně nebo za užití kontroléru) provoz ve stroboskopickém režimu, aby bylo možno osvětlení triggerovat z kamery a neprovozovat jej ve spojitém režimu, což mj. prodlouží životnost osvětlovače. Musí se jednat o monochromatický osvětlovač pro eliminaci chromatických vad. Chromatické vady se projevují zejména na přechodech mezi tmavými a světlými částmi obrazu (viz. Vady objektivu v kapitole 2.2), způsobují na těchto přechodech rozmazání monochromatického obrazu a tím snížení přesnosti.

## Kamera

Kamera musí disponovat dostatečným rozlišením, aby byla schopna zachytit veškeré vady i na součástkách velikosti 0805. Pixely na senzoru kamery by měly být co největší, aby byly schopny zachytit co nejvíce světla a tím minimalizovat expoziční dobu. Kamera by také měla disponovat senzorem co nejnovější generace, protože obraz z těchto senzorů vzhledem k velmi rychlému vývoji v oblasti strojového vidění obsahuje i při vyšších hodnotách zesílení mnohem méně šumu, než obraz ze starších senzorů.

## Expozice

Expoziční čas byl určen následovným výpočtem:

$$\text{Délka pásu} = 1500 \text{ mm} \tag{15}$$

$$\text{Doba jednoho oběhu pásu} = 15 \text{ s}$$

$$\text{Rychlost pásu} = \frac{1500 \text{ mm}}{15 \text{ s}} = 100 \text{ mm/s}$$

$$\text{Změřené rozlišení kamery} = 3,72 \text{ } \mu\text{m/px}$$

$$\begin{aligned} \text{Čas, za který ujede součástka vzdálenost odpovídající jednomu pixelu} &= \frac{3,72 \text{ } \mu\text{m/px}}{100 \text{ mm/s}} \\ &= 37,2 \text{ } \mu\text{s/px} \end{aligned}$$

## Objektiv

Objektiv musí mít dostatečnou rozlišovací schopnost, aby příliš nedegradoval rozlišení kamery, taktéž musí být určen pro velikost čipu, který bude mít zvolená kamera. Zvětšení objektivu musí být takové, aby byla součástka v obraze dostatečně velká, ale zároveň aby zorné pole poskytovalo dostatečnou toleranci pro posuv součástky v obou směrech. Světelnost objektivu by měla být co největší, aby umožnil co nejkratší expoziční čas. Hloubka ostroty musí být dostatečná, aby pokryla rozptýl rozměrů jedné výrobní řady součástek (výšky pro horní kameru a šířky pro boční kameru), pro různé rozměry je využito stavění mikroposuvem. Nastavitelná clona může být výhodou, ale není nutností, pokud má objektiv s fixní clonou dostatečnou hloubku ostroty. Objektiv nemusí mít ostřicí kroužek – lze ošetřit mikroposuvem.

Kamery s objektivy musí být umístěny na mikroposuvech umožňujících stavění alespoň ve dvou osách (v příčném směru k pásu a v ose směrem od a k pásu), třetí osu je možno upravit zpožděním generování triggeru pro kameru. Velmi důležitým kritériem pro všechny HW součásti je také cena.

### 6.2.2 Software

SW inspekce by měl být dostatečně robustní, aby bylo možno využít jeden algoritmus pro boční i horní pohled, pro obě velikosti součástek, pro oba typy těl součástek, a tento algoritmus

jen nastavovat vnějšími parametry. Zvláštní důraz musí být kladen na jednoduchost a rychlost algoritmu, aby bylo možno docílit žádaného taktu součástek. Jednotlivé části algoritmu musí být v maximální míře paralelizovatelné. Celý algoritmus musí umožňovat běh ve 4 instancích, aby bylo možno paralelně vyhodnocovat obraz ze všech čtyř kamer.

### 6.3 Návrh řešení kamerové inspekce

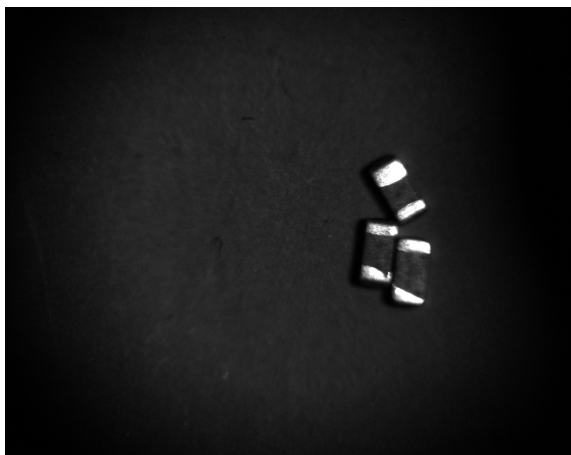
Návrh řešení probíhal na základě dostupných vzorků vadných kusů s popisem vad (např. chybějící terminace, zrnko v těle atp.) především experimentální cestou (testování konkrétního HW a vad). Během testů se ukázala jako nejvhodnější níže popsaná sestava.

#### 6.3.1 Hardware

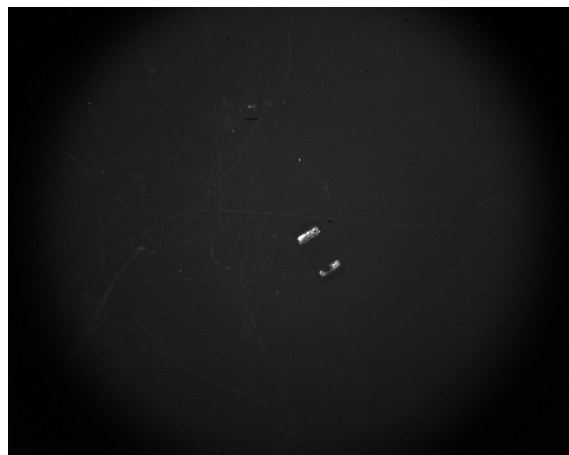
##### Osvětlení

Postup výběru **osvětlení** byl následující: nejprve byl určen typ osvětlovače. Vzhledem k tomu, že kontrolované součástky jsou velmi malé, byl zvolen bodový osvětlovač. Vzhledem k nemožnosti objektivně porovnat osvětlovače pouze na základě dostupných parametrů udávaných výrobcem byla provedena série testů s různými osvětlovači i objektivy. Jako velmi výhodné zejména s ohledem na prostorové požadavky se jevilo užití objektivu s polopropustným zrcadlem, který umožňuje přes jednu optickou soustavu na součástku nahlížet a zároveň i svítit, protože obsahuje boční otvor pro umístění bodového světla. Bohužel, žádný z dostupných objektivů nebyl vhodný svým zvětšením a zejména světelností (polopropustné zrcadlo snižuje intenzitu dopadajícího světla), a tak bylo nutno nastavit neúměrně vysoké expoziční časy.

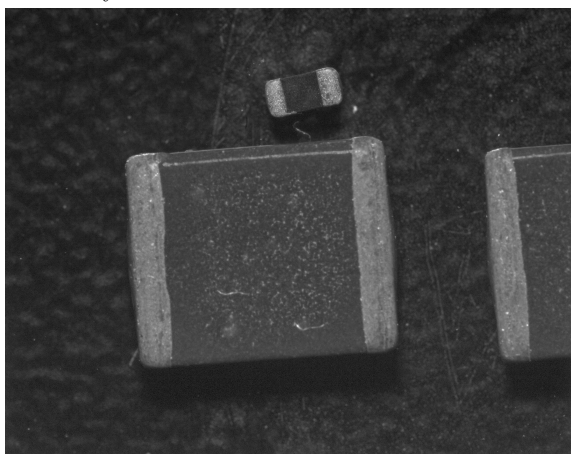
Na snímcích níže uvádím několik snímků z testování různých objektivů a osvětlovačů. Obr. 20a zobrazuje test běžného objektivu s mezikroužky. Obraz je ostrý prakticky pouze ve svém středu, směrem ke krajům je velmi rozostřený, na krajích je patrná silná vinětace. Obr. 20b byl pořízen během testu s objektivem s polopropustným zrcadlem a koaxiálním osvětlením. Kromě toho, že objektiv nebyl určen pro danou velikost čipu (viz ořez snímku), koaxiální osvětlení svým charakterem nezvýraznilo některé vady, ani kontury samotné součástky. Obr. 20c zobrazuje snímek při polarizovaném osvětlení. Terminace jsou sice dobře detekovatelné, ale vady na těle (v tomto případě drobné bubliny ve skleněné vrstvě) nejsou viditelné. Poslední obr. 20d zobrazuje snímek s finálně vybraným osvětlovačem CCS HLV3-22RD-4S.



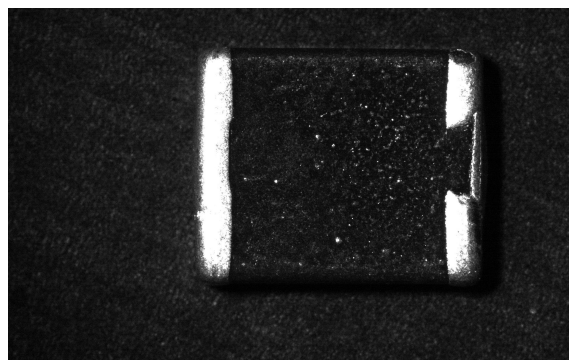
(a) Testovací snímek s běžným objektivem s mezikroužky



(b) Testovací snímek s koaxiálním osvětlením



(c) Testovací snímek s polarizovaným osvětlením



(d) Testovací snímek s finálním osvětlovačem

Obr. 20: Příklady snímků z testování HW

Na konci těchto testů byly vytipovány dva osvětlovače splňující výše uvedené požadavky. Jednalo se o osvětlovač Effilux EFFI-SHARP-FAN-MX2 a CCS HLV3-22RD-4S (obr. 21b). Výhodou osvětlovače Effilux byl mírně vyšší průměrný jas v obraze při stejných expozičních parametrech. Nevýhodou ovšem byl řádově větší elektrický příkon (3,2 W u CCS vs. 90 W u Effilux) a mnohem větší rozměry (bez optiky CCS 22 x 46.6 x 49 a Effilux 151 x 79 x 79 mm), hmotnost a hlučnost, vzhledem k nutnosti aktivního chlazení ventilátorem. Jelikož výsledky s osvětlovačem CCS byly akceptovatelné, byl pro řešení vybrán právě tento osvětlovač. Pro řešení úlohy byly nutné 4 osvětlovače (2x pro horní, 2x pro boční pohled), na 3 z nich je využit s doporučeným objektivem HL2-25-P, na čtvrtém je z prostorových důvodů osvětlovač bez objektivu. Osvětlovače jsou zapojeny do kontroléru CCS PJ2-3005-4CA-PE (obr. 21b), který umožňuje mj. triggerovat jednotlivé osvětlovače přes digitální vstupy.

Tab. 1: Základní parametry zvoleného osvětlovače  
Převzato z [14]

<b>Typ</b>	CCS HLV3-22RD-4S
<b>Vlnová délka</b>	630 nm
<b>Příkon</b>	3,2 W
<b>Rozměry</b>	22 x 46,6 x 49 (Š x H x V)
<b>Hmotnost</b>	53 g
<b>Maximální proud</b>	1000 mA



(a) Osvětlovač CCS HLV3-22RD-4S



(b) Kontrolér CCS PJ2-3005-4CA-PE

Obr. 21: Snímek zvoleného osvětlovače a kontroléru  
Převzato z [14]

## Kamera

Kamery byly vybírány z osvědčené výrobní řady ace firmy Basler. Tyto kamery se vyznačují mj. svými kompaktními rozměry, kvalitními senzory s dobrými obrazovými parametry a dostatečným rozlišením, podporou PoE v GigE verzi, možností triggerování digitálním vstupem a možností využití digitálního výstupu (ten lze měnit např. na základě časovače nebo při expozici). Sběrnice byla zvolena GigE, napájení přes PoE, což bylo výhodné vzhledem ke zvolené DIO kartě (viz dále, 2x NI-8237R s 2x Ethernet portem s podporou PoE). Jako nejvhodnější se ukázala kamera acA2040-35gm, která disponuje dostatečným rozlišením (3 MP, 2048 x 1536 px), dostatečnou snímkovou frekvencí (36 FPS), senzorem poslední generace s dobrou obrazovou kvalitou (Sony IMX265) a velkými pixely (3,45 x 3,45  $\mu\text{m}$ ).

Tab. 2: Základní parametry zvolené kamery  
Převzato z [15]

<b>Typ</b>	Basler acA2040-35gm
<b>Senzor</b>	Sony IMX 265
<b>Rozměr senzoru</b>	1/1,8"
<b>Rozlišení senzoru</b>	2048 x 1536 px
<b>Velikost pixelu</b>	3,45 $\mu\text{m}$ x 3,45 $\mu\text{m}$
<b>Snímková frekvence</b>	36 FPS
<b>Komunikační rozhraní</b>	GigE



Obr. 22: Kamera Basler ace GigE

Kamera je triggerována hardwarově přes digitální vstup náběžnou hranou z FPGA. Expoziční doba je nastavena na 120  $\mu\text{s}$ . Po celou dobu expozice je zároveň držena logická „1“ na digitálním výstupu kamery, který přes kontrolér CCS triggeruje osvětlovač příslušný ke kameře. Expoziční čas byl experimentálně upraven tak, aby došlo k dostatečnému osvětlení součástky bez jejího rozmazání.

## Objektiv

Vzhledem k charakteru úlohy byly zvažovány pouze dvě kategorie objektivů, a to telecentrické a makroobjektivy. Telecentrický objektiv by byl vhodný zejména pro přesné měření rozměrů, ovšem proti jeho využití hovořily jejich velké rozměry v kombinaci s velkou pracovní vzdáleností, malá hloubka ostrosti a vysoká cena.

Vzhledem k dostupným objektivům, velikosti senzoru kamery a velikosti součástek a potřebné hloubce ostrosti byl jako optimální určen objektiv se zvětšením 1x. Z nabídky objektivů byl vybrán objektiv VS Technology VS-MC1-40, jedná se o makroobjektiv s požadovaným zvětšením, dostačujícím rozlišením a zejména kompaktními rozměry. Experimentem bylo zjištěno skutečné zvětšení 1,07x.

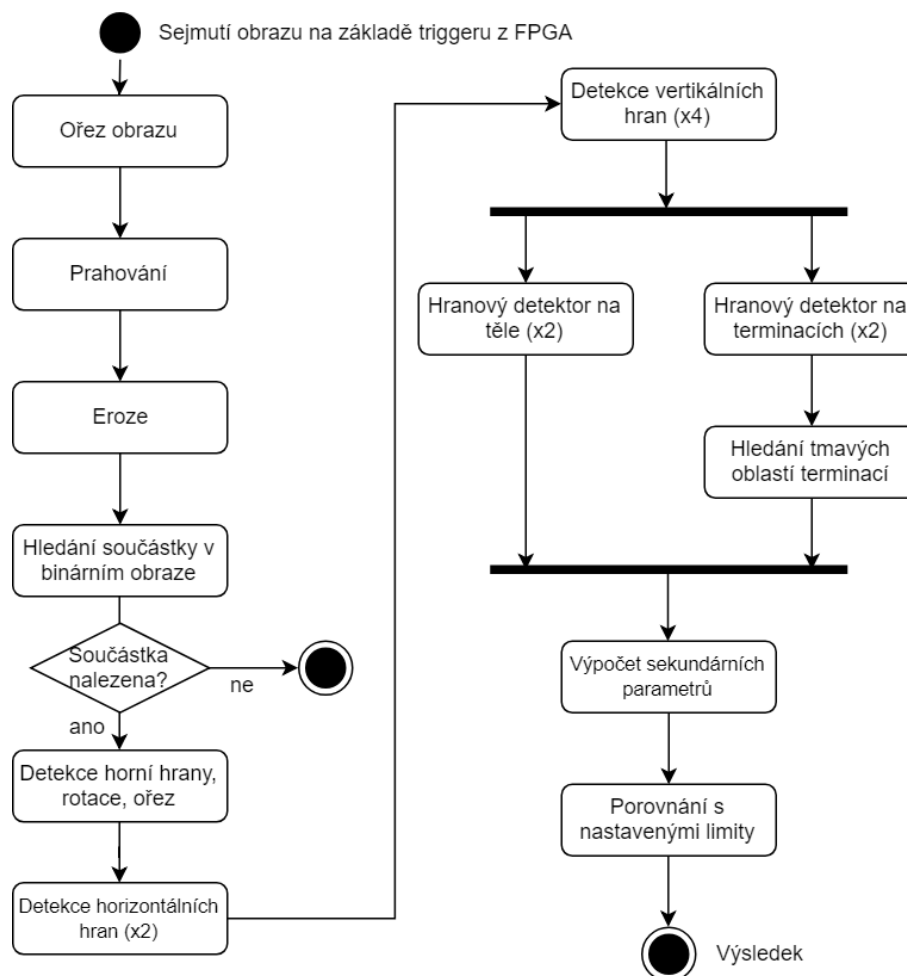
Tab. 3: Základní parametry zvoleného objektivu  
Převzato z [16]

<b>Typ</b>	VS-MC1-40
<b>Zvětšení</b>	1x
<b>Pracovní vzdálenost</b>	41,7 mm $\pm$ 2 mm
<b>Světelnost</b>	f/6,58
<b>Rozměry</b>	52,7 x 30 x 30 mm
<b>Maximální velikost senzoru</b>	2/3"

### 6.3.2 Software

Celý algoritmus a v něm užití funkce byly navrženy a implementovány se zvláštním důrazem na rychlost, aby bylo možno docílit žádaného taktu kontroly. Veškeré vytvořené subVI, které jsou součástí algoritmu pro detekci a vyhodnocení součástky mají nastaven mód „inline“, tj. nejsou volány jako samostatné funkce, nýbrž jsou během kompilace vloženy do těla nadřazeného VI. Byly vypuštěny složité, pomalé funkce (např. hledání vzorů, porovnání kontur, konvoluce) ve prospěch jednodušších a rychlejších (prahování, hledání hran). Algoritmus byl vytvořen jako reentrantní, tj. s možností vícenásobného paralelního volání (dohromady 4x, pro každou kameru zvlášť).

Pro prvotní testy a návrhy algoritmu strojového vidění byl využit SW VBAI, kde bylo možno jednotlivé kroky úpravy a zpracování obrazu odzkoušet mnohem rychleji, než by bylo možné za užití LabVIEW. Poté došlo k implementaci LabVIEW za současného testování výkonnosti algoritmu. Algoritmus byl postupně doplněn či modifikován dle požadavků zákazníka či vzhledem k nutnosti dodržet požadovaný takt výroby. Při testování algoritmu bylo sledováno využití CPU – cílový stav využití CPU při taktu  $< 8$  součástek za sekundu bylo 75 %, což se podařilo dodržet, a délka zpracování jednoho snímku do 60 ms, což se taktéž podařilo dodržet.



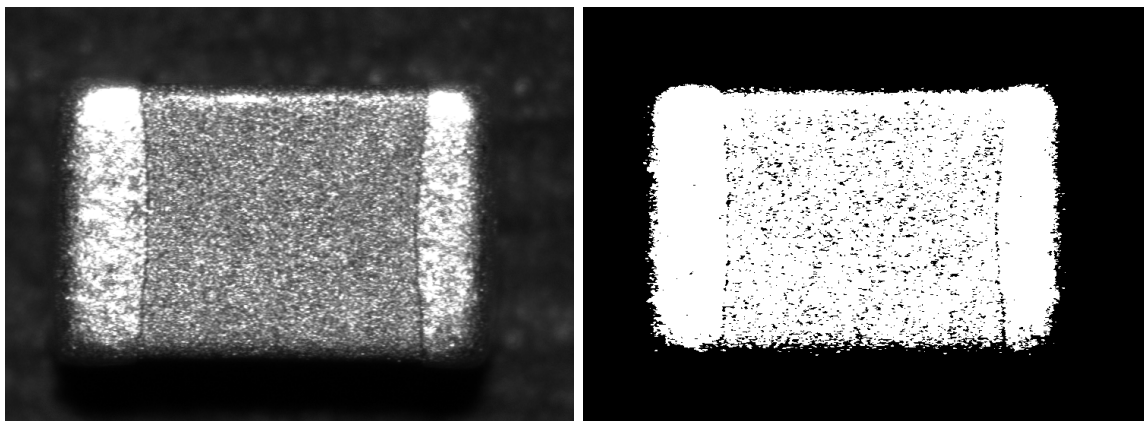
Obr. 23: Diagram popisující průběh zpracování a vyhodnocení obrazu

### 6.3.3 Popis algoritmu

Algoritmus probíhá následovně (viz obr. 23):

1. **Ořez obrazu z kamery** – zorné pole je kamery je větší, než je samotná součástka. Součástka se v zorném poli kamery díky přesnosti zarovnání pozice za pomoci kovových koleček a vodiček příliš nepohybuje, proto je možno obraz příčně ke směru jízdy pásu ořezat, aby následující kroky probíhaly pouze na části obrazu, kde se součástka s největší pravděpodobností nachází.
2. **Prahování** – pro detekci součástky v obraze je využita série poměrně rychlých kroků (v porovnání např. s hledáním vzorů). Prvním z nich je prahování, kdy je obraz převeden z monochromatické do binární podoby. Použito je globální prahování, tj. všechny pixely původního obrazu v určitém intervalu hodnot (threshold range) mají v binárním snímku hodnotu „1“, všechny mimo tento rozsah zase „0“. Příklad snímku před aplikací operace prahování je na obr. 24a, po prahování na obr. 24b.





(a) Snímek před prahováním

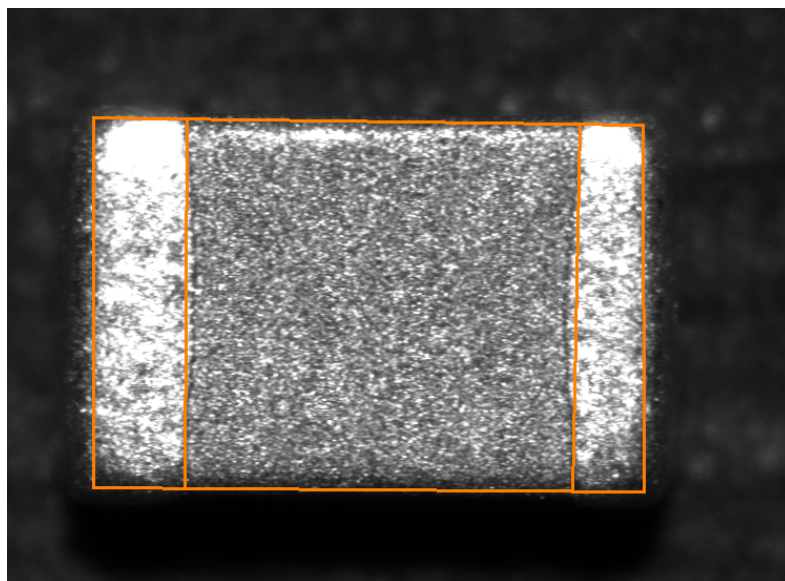
(b) Snímek po prahování

Obr. 24: Příklad binárního snímku součástky

3. **Eroze/dilatace** – protože globální práh nelze nastavit tak, aby byla zachycena v binárním obraze celá součástka bez rušivého okolí (např. nečistoty na páse), je možnost v algoritmu na základě nastavení příslušného parametru v binárním obraze provést jeho erozi nebo dilataci. Zjednodušeně řečeno – oba algoritmy upravují velikost oblastí s hodnotou „1“. Eroze tyto oblasti zmenšuje o několik pixelů na základě počtu iterací a zvoleného jádra, dilatace je naopak zvětšuje. Dilataci je možno nastavit pro zahmlení „děr“ – izolovaných oblastí s hodnotou „0“ v oblastech s hodnotou „1“. Erozi je naopak možno nastavit pro eliminaci malých oblastí o hodnotě „1“ (např. nečistoty na páse).
4. **Hledání součástky v binárním obraze** – probíhá za pomoci funkce *IMAQ Count Objects* z VDM, která umožňuje vyhledávání objektů v monochromatickém obraze. V případě, že součástka na základě daných kritérií není nalezena, zbylá část algoritmu se již neprovede.
5. **Detekce horní hrany, rotace, ořez** – Protože součástka může být v obraze mírně natočena, detekuje se její horní hrana. ROI pro hledání horní hrany je vztaženo ke středu nalezeného objektu z předchozího kroku. Je určen úhel natočení horní hrany a následně je obraz otočen a opětovně ořezán. Otočení umožňuje jednoduše využít funkce VDM, které jsou optimalizovány pro hledání vertikálních či horizontálních hran bez natočení. Obraz je poté znovu ořezán, ze stejného důvodu jako výše.
6. **Detekce horizontálních hran** – poté, co je k dispozici snímek, který s velkou pravděpodobností obsahuje kontrolovanou součástku, probíhá její přesná lokalizace. Ta začíná hledáním horizontálních hran, tj. horní a dolní. Hledání probíhá na základě nastavitelných parametrů, v ROI vztaženém ke středu součástky, za pomoci funkce VDM *IMAQ Rake*. Lze nastavit, zda bude probíhat hledání shora dolů nebo zdola nahoru, zda se hledají náběžné nebo sestupné hrany a v jakém intervalu a parametry hledání hran (kernel, šířka, síla).

Z nalezených dílčích hran je provedeno proložení přímkou. Výsledkem této funkce je úsečka představující horní, resp. dolní hranu součástky. Hledání obou hran probíhá paralelně.

7. **Detekce vertikálních hran** – probíhá s mírnými odlišnostmi shodně jako u horizontálních hran. Vyhledávání probíhá zprava doleva nebo zleva doprava. Výsledkem jsou úsečky představující levou, resp. pravou hranu a hranu levé, resp. pravé terminace. Vyhledávání opět probíhá paralelně. Po proběhnutí této funkce je k dispozici přesně lokalizovaná součástka, viz obr. 25.



Obr. 25: Lokalizovaná součástka

8. **Hranová detekce vad** – pro detekci vad je využito hranového detektoru. Tato detekce probíhá jak na těle součástky, tak na terminacích. Pro tyto detekce je využita programově vypočítaná osnova, po které detekce hran probíhá, opět paralelně.
  - **Hranový detektor na těle** – probíhá dvakrát, jednou pro původní obraz, čímž jsou lokalizovány zejména světlé vady, podruhé na filtrovaném obraze (morfologie - dilatace), čímž jsou zachyceny tmavé vady.
  - **Hranový detektor na terminacích** – probíhá pro každou terminaci zvlášť.
9. **Hledání tmavých oblastí terminací** – jelikož se během testování ukázalo, že hranový detektor neumí detekovat všechny tmavé vady terminací, bylo doplněno hledání tmavých oblastí terminací. Hledání funguje na principu prahování na základě programově vypočtené hodnoty prahu. Výsledkem je poměr tmavých pixelů v procentech, který určuje velikost tmavé oblasti.
10. **Výpočet sekundárních parametrů** – poté, co jsou provedeny veškeré detekce v obraze, jsou vypočteny z dostupných dat sekundární parametry součástky. Jedná se o rozměry

součástky a terminací ze známých pozic jejích hran a o výpočet úhlů mezi jednotlivými hranami (určuje, zda součástka např. nemá ulomený roh).

11. **Porovnání s nastavenými limity** – každý ze změřených a vypočítaných parametrů je porovnán s limity. Některé limity lze vypnout, v takovém případě pak nemají vliv na označení součástky za bezvadnou/vadnou. Na základě dílčích výsledků je určen celkový výsledek, a to v případě, že kterýkoliv z dílčích výsledků je mimo limity, je celkově součástka označena za vadnou.

#### 6.3.4 Nastavení algoritmu

Protože výše zmíněný algoritmus je univerzální pro boční i horní pohled a pro všechny kontrolované součástky, je nutno jej parametrizovat, tj. upravovat vyhledávací parametry jednotlivých funkcí algoritmu. Parametrizovat lze např. nastavení zpracování binárního obrazu (prahování, eroze/dilatace, minimální velikost nalezeného objektu), veškeré nastavení vyhledávání hran (polarita hrany, vyhledávací jádro, šířka, síla, směr vyhledávání), vyhledávací oblasti pro jednotlivé kroky. Nepředpokládá se, že by koncový uživatel testeru tyto parametry často měnil, jsou v aplikaci k dispozici zejména pro počáteční doladění či změnu v případě problematické detekce.

Tab. 4: Kontrolované parametry

Kategorie	Parametr	Jednotka	Popis
<b>Rozměry</b> (přepínatelná jednotka měření)			
	Rozměr X	px nebo mm	Rozměr v ose X obrazu
	Rozměr Y	px nebo mm	Rozměr v ose Y obrazu
	Šířka L/P terminace	px nebo mm	Rozdíl mezi šířkami obou terminací
	Rozdíl terminací	px nebo mm	
	Délka těla	px nebo mm	
<b>Úhly</b>			
	Úhel mezi hranami (x5)	°	Porovnání úhlu mezi jednotlivými proloženými linkami (horní a postupně všechny ostatní linky)
<b>Hrany</b>			
	Kvalita hrany (x6)	-	Maximální odchylka mezi nalezenou hranou a proloženou linkou
	Podíl nalezených hran (x6)	%	Podíl nalezených hran, které se účastní na proložení linky
<b>Tělo</b>			
	Minimální hodnota	-	Minimální intenzita v obraze
	Maximální hodnota	-	Maximální intenzita v obraze
	Střední hodnota	-	Střední intenzita v obraze
	Směrodatná odchylka	-	Směrodatná odchylka intenzity v obraze
	Počet hran	-	Počet nalezených hran na těle (= vady)
<b>L/P terminace</b>			
	Minimum	-	Minimální intenzita v obraze
	Maximum	-	Maximální intenzita v obraze
	Střední hodnota	-	Střední intenzita v obraze
	Směrodatná odchylka	-	Směrodatná odchylka intenzity v obraze
	Počet hran	-	Počet nalezených hran na těle (=vady)
	Poměr světlých pixelů	%	Poměr počtu pixelů majících intenzitu vyšší, než programově vypočítaný práh

## 7 Návrh řízení testeru

### 7.1 Požadavky na řízení

Během přípravné fáze projektu byly definovány následující požadavky na řízení stroje:

1. Vibrační podavač, DC motor pásu je nutno zapnout po spuštění inspekce, a naopak vypnout po jejím zastavení.
2. Řídicí systém stroje musí obdržet informaci o přítomnosti součástky na pásu. K tomuto bude využito optické přítomnostní čidlo firmy Keyence.
3. Zpoždění mezi detekcí součástky a pořízení snímku kamerou musí být konfigurovatelné, stejně tak zpoždění mezi detekcí součástky a vzduchovým ventilem s tryskou. Toto zpoždění bude odvozeno od pozice pásu, nikoliv od času.
4. Při otevření dveří nebo při nedostatečném tlaku vzduchu je zastaven motor a vibrační podavač. Otevření dveří navíc iniciuje rozsvícení vnitřního osvětlení.
5. Při stisknutí tlačítka STOP je kromě zastavení motoru a podavače vypnut také hlavní vzduchový ventil.
6. Vzduchové ventily s tryskou budou ve výchozím stavu otevřené, aby došlo ke sfouknutí všech součástek, u kterých nedošlo k vyhodnocení obrazu (např. nebyly zachyceny přítomnostním čidlem).

### 7.2 Návrh řízení testeru

Informace o směru pohybu a pozici pásu je zajištěna čítačem pulsů od kvadraturního enkodéru. Tento enkodér musí mít takovou přesnost, aby posun pásu o délku součástky rozměru 0805 (cca 2 mm) vyvolal na výstupu enkodéru řádově několik jednotek až desítek pulzů. Pozici pásu není nutno znát absolutně, postačuje relativní informace. Čítač čítá vpřed i vzad na základě náběžných hran a úrovní obou kanálů A i B od kvadraturního enkodéru.

Pro realizaci algoritmu řízení testeru jsou využity 2 DIO karty NI-8237R. Každá tato karta má 8 izolovaných digitálních vstupů a 8 výstupů, 2 TTL konfigurovatelné vstupy/výstupy, 2 diferenční kanály pro připojení kvadraturního enkodéru a 2 GigE porty s podporou PoE. Vstupy a výstupy jsou interně na kartě připojeny na FPGA čip Xilinx Spartan 6, na němž se implementuje žádaná funkcionalita mezi vstupy a výstupy. Užití těchto karet pro řízení testeru je z mnoha pohledů velmi výhodné – veškeré vstupy a výstupy pro řízení testeru jsou pouze digitální (dvoustavové), takže plně postačují pouze digitální vstupy a výstupy, diferenční vstupy umožňují připojení enkodéru od pásu, GigE porty umožňují napájení kamer a jejich připojení do PC a FPGA čip umožňuje žádané přesné časování (pro čítání enkodéru, generování triggerů kamery atp.). Taktovací frekvence FPGA je 40 MHz, FPGA neumožňuje zpracování obrazu ani implementaci FIFO mezi FPGA a PC.

Základní úroveň řízení je implementována přímo na FPGA. Část řízení nevhodná svou složitostí pro FPGA, případně závisující na jiné části SW, je implementována na PC. DIO karty s FPGA jsou vzhledem k počtu vstupů a výstupů využity obě – jedna pro rychlé signály, druhá pro ostatní.

**FPGA karta pro rychlé signály** čítá náběžné hrany z enkodéru, čítač čítá vpřed i vzad. Taktéž tato karta zaznamenávat náběžné hrany od přítomnostního čidla. Na základě náběžné hrany a hodnoty čítače generuje s konfigurovatelným zpožděním triggeru kamery a pulz pro vzduchový ventil.

Vzhledem k omezeným hardwarovým prostředkům dostupných na čipu FPGA je nutno proměnnou čítače implementovat jako nejmenší možný datový typ, aby nepřetékal příliš často (tj. minimálně po jedné otočce pásu).

Funkcionalita vztažená ke kontrolní stanici je na kartě duplikována, v rámci bloků pro jednu kontrolní stanice je duplikována funkcionalita vztažená ke kameře.

**FPGA karta pro pomalé signály** zajišťuje propojení fyzických vstupů a výstupů do vyšších vrstev SW. Kromě toho je na ni implementována logika při otevření dveří, nízkém tlaku a stisknutí tlačítka STOP.

### 7.3 Realizace logiky řízení na kartě pro rychlé signály

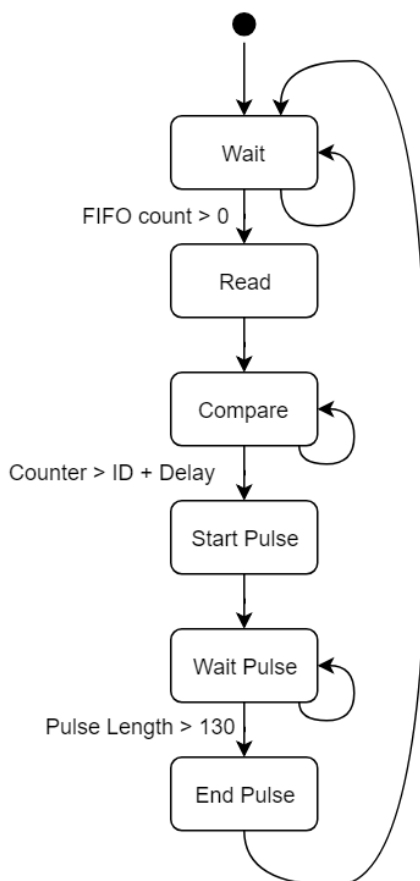
Zjednodušeně je logika řízení na FPGA pro rychlé signály popsána na obr. 26. Modrou barvou jsou vyznačeny fyzické vstupy a výstupy. Zelenou barvou jsou zobrazeny vstupy a výstupy z nadřazené aplikace na PC. Šedé bloky jsou bloky FIFO, které jsou součástí knihovny NI FPGA. Přerušovanou čarou jsou ohraničeny bloky, které jsou na FPGA duplikovány. Jednoduchou šipkou jsou označeny 1bitové signály, dvojitou šipkou jsou označeny vícebitové signály. Vnější blok náleží každé kontrolní stanici, tj. má dvě instance. V něm vnořený blok náleží kameře, proto je dvakrát v každé stanici, tj. dohromady čtyřikrát pro celou aplikaci.

Prvním blokem je **čítač**. Jedná se o čítač vpřed i vzad od hodnoty  $-32768$  do  $32767$ , slouží pro určování relativní polohy pásu. Jeho vstupy jsou kanály A a B od kvadraturního enkodéru na páse. Datový typ čítače je I16, což je nejmenší datový typ, který přeteče po více než jednom kompletním oběhu pásu.

Následuje blok **detekce přítomnosti**, který slouží k zachycení náběžné hrany od přítomnostního čidla. Tento blok je zdvojen, pro každou kontrolní stanici je jeden. Jako ochrana proti součástkám, které by mohly jet po páse příliš blízko za sebou, proti tzv. „přetriggerování“ kamery (kamera obdrží dva nebo více triggerovacích pulzů za sebou v menším časovém rozmezí, než je ve kterém je schopna snímat další snímek), či proti zákmitům na výstupu přítomnostního čidla, je implementováno zpoždění, kdy po první zachycené náběžné hraně je další ignorována do doby, než pás ujede přednastavenou vzdálenost. V případě, že je náběžná hrana od přítomnostního čidla zachycena, je uložena aktuální hodnota čítače do třech pamětí FIFO. Hodnota čítače



slouží jako ID detekované součástky, ale zároveň určuje, v jaké poloze na páse se součástka nachází. Tyto tři paměti FIFO jsou SC\_FIFO sloužící pro blok triggeru kamery, ID\_FIFO sloužící pro nadřazenou aplikaci v PC a ID\_INT\_FIFO sloužící pro blok pulzu ventilu.



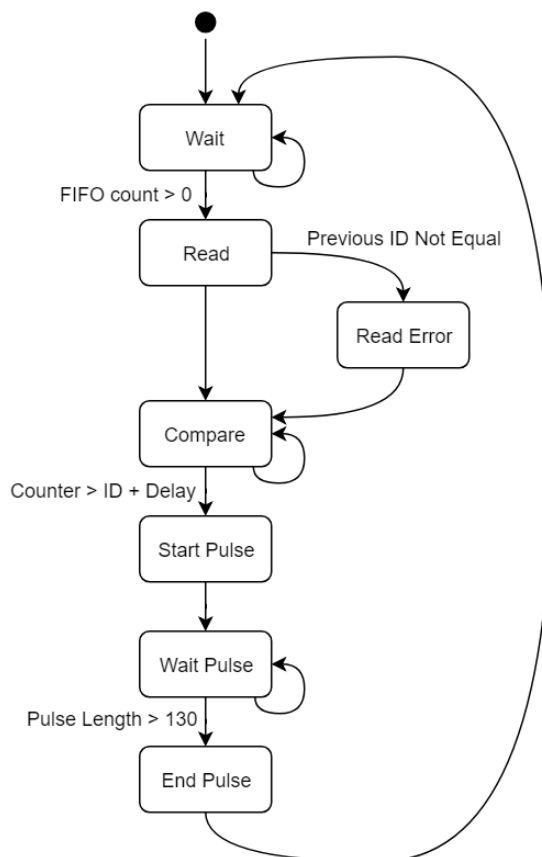
Obr. 27: Stavový diagram bloku trigger kamery

Blok **trigger kamery** slouží pro generování HW triggeru pro digitální vstup kamery, je implementován jako stavový automat, viz obr. 27. Tento blok vždy čeká na naplnění paměti SC\_FIFO (stav *Wait*). Pokud je v ní k dispozici alespoň jeden prvek, vyčte jej (*Read*). Následně porovnává, zda je součet aktuální hodnoty čítače a z PC definovaného zpoždění CAM\_DELAY (v počtech pulzů, udává tedy vzdálenost mezi přítomnostním čidlem a kamerou v pulzech) větší než vyčtené ID z paměti (tj. poloha součástky na páse, *Compare*). Pokud součástka dosáhla žádané polohy pod kamerou, tj. výše zmíněná hodnota čítače je vyšší, generuje se trigger pro kameru (*Start Pulse*). Zároveň je ID součástky uloženo do paměti CN\_FIFO, ve které uloženy všechna ID součástek, které už byly nasnímány, ale ještě nebyly vyhodnoceny/vyřazeny z pásu. Poté j pulz je po fixně dané době ukončen (*End Pulse*). Tento blok se na FPGA nachází 4x, pro každou kameru zvlášť.

Blok **pulz ventilu** zajišťuje generování pulzu pro vzduchový ventil, sloužící pro sfouknutí vadné součástky z pásu. Blok zajišťuje, aby byl ventil ve výchozím stavu trvale otevřen, takže



v případě chyby nadřazeného SW, příliš dlouhého vyhodnocování snímku, nedetekování součástky atp., jsou součástky sfoukávány mezi vadné kusy, což je „bezpečné“ chování dle požadavků zákazníka. Tento blok, podobně jako blok pro trigger kamery, je implementován jako stavový automat, viz obr. 28. Čeká na naplnění paměti CN\_FIFO (stav *Wait*), pokud je v ní alespoň jeden prvek, vyčte jej (*Read*), poté čeká, dokud součet aktuální hodnoty čítače a z PC definovaného zpoždění NOZZLE\_DELAY (v počtech pulzů, udává tedy vzdálenost mezi přítomnostním čidlem a vzduchovou tryskou v pulzech) větší než vyčtené ID z paměti (tj. poloha součástky na páse, *Compare*). Pokud součástka dosáhne žádané polohy u trysky, vyčítá se ID\_RES\_FIFO, RES\_FIFO a ID\_INT\_FIFO. Porovnává se, zda je shodné ID z ID\_RES\_FIFO, tj. z nadřazené aplikace v PC a z FPGA, což má za úkol odhalit, zda aktuálně dostupný výsledek náleží k právě projíždějící součástce (tj. zda nedošlo z nějakého důvodu ke ztrátě snímku z kamery, pozdnímu zpracování obrazu atp., *Start Pulse*). Pokud ID nejsou shodná, nedojde k vypnutí ventilu a součástka končí ve vadných kusech. Pokud ID jsou shodná, dojde k případnému vypnutí ventilu na základě výsledku z RES\_FIFO. Doba vypnutí ventilu má délku NOZZLE\_LENGTH, která je opět odvozena od počtu pulzů enkodéru (*Wait Pulse*), takže i v případě kolísání rychlosti pásu dojde ke korektnímu průjezdu/sfouknutí součástky. V případě, že došlo k neshodě v ID při předchozím průjezdu součástky, je tato chyba ošetřena ve stavu *Read Error* vyčtením chybných ID z FIFO.



Obr. 28: Stavový diagram bloku pulz ventilu

#### 7.4 Realizace logiky řízení na kartě pro pomalé signály

DIO karta s FPGA pro pomalé signály zajišťuje návaznost mezi ostatními signály stroje. FPGA generuje výstup pro signalizační maják na základě hodnoty enumerátoru z PC. V případě, že je nedostatečný tlak vzduchu nebo jsou otevřeny dveře mimo servisní mód, je signalizován chybový stav (červené světlo na majáku) a je vypnut motor a vibrační podavač. Při stisknutí tlačítka „STOP“ se navíc vypíná i hlavní vzduchový ventil. Pro výstup motoru je navíc k dispozici PWM o frekvenci 400 Hz s nastavitelnou střídou. Pro signalizaci stroje je instalován světelný maják.

Jelikož je FPGA do jisté míry nezávislé na nadřazené aplikaci na PC (v případě neočekávaného pádu aplikace na PC FPGA běží dále), je na FPGA implementován jednoduchý watchdog. Spočívá v postupné inkrementaci hodnoty proměnné FPGA v PC. V případě, že PC do daného časového úseku od poslední inkrementace opět hodnotu neinkrementuje, FPGA uvede všechny výstupy do bezpečného stavu, tj. vypnuto.

## 8 Návrh statistického zpracování výsledných dat

Data z kontrolovaných komponent jsou statisticky zpracovávána dvojím způsobem:

- Pro účely provozního přehledu o výrobě jsou průběžně vypočítávány parametry počtu zkontrolovaných součástek, počtu bezvadných a vadných kusů, výtěžnosti, počty vadných kusů dle jednotlivých kategorií, taktu součástek v čase.
- Pro účely navržení parametrů z bufferu snímků jsou počítány průměry, směrodatné odchylky jednotlivých parametrů na základě nichž jsou navrhovány limity.

### 8.1 Provozní statistika

Provozní statistika je počítána z kontrolovaných součástek průběžně při chodu stroje. Nejdůležitější údaje se zobrazují na hlavním provozním panelu aplikace, detailnější jsou k dispozici v konfigurační části. Na základě požadavků provozní statistika zahrnuje následující parametry:

1. **Počet zkontrolovaných kusů v jednotlivých stanicích**, počítáno na FPGA z počtu náběžných hran od čidla přítomnosti.
2. **Počet zkontrolovaných kusů každou kamerou**, počítáno z počtu snímků sejmutých každou kamerou.
3. **Počet bezvadných kusů**.
4. **Počet vadných kusů** a počet v jednotlivých kategoriích výsledků, ve kterých byl kus označen za vadný.
5. **Takt kontrol**, počítaný na PC z počtu náběžných hran od čidla přítomnosti za časovou jednotku. Zobrazován i jako graf v čase za posledních cca 30 minut.
6. **Výtěžnost** celková a jednotlivých stanic, tj. procento bezvadných kusů z celkového počtu. Zobrazována i jako graf v čase za posledních cca 30 minut.

Veškeré statistiky může uživatel mazat. Po ukončení aplikace se aktuální statistiky ukládají do textového souboru, aby byly k dispozici po opětovném spuštění stroje, nedochází ale k jejich archivaci např. do databáze.

### 8.2 Návrh limitů parametrů optické inspekce

Tato funkce je navržena pro implementaci v konfigurační části aplikace, konkrétně v pluginu pro konfiguraci parametrů a limitů. Před zpracováním je nutno nasnímat buffer snímků konkrétní kamery (ať už přímo z kamery nebo načtením z disku). Následně je možno v bufferu odznačit snímky, které si uživatel nepřeje zahrnout do návrhu parametrů (např. protože mají vadu). Poté je po stisku příslušného ovládacího tlačítka provedena analýza všech označených snímků v bufferu. Z výsledků jsou vypočteny průměr, minimum, maximum a směrodatná odchylka každého

parametru, též je zobrazen počet bezvadných a vadných kusů při aktuálním nastavení parametrů. Poté probíhá návrh nových limitů jednotlivých parametrů, a to skrz dialogové okno, kde uživatel zadá procentuální hodnotu dolního a horního limitu, která je odvozena od směrodatné odchylky. Limity se tedy vypočtou:

$$\text{horní limit} = \text{průměr} + \text{směrodatná odchylka} \cdot \frac{\text{zadané procento}}{100} \quad (16)$$

$$\text{dolní limit} = \text{průměr} - \text{směrodatná odchylka} \cdot \frac{\text{zadané procento}}{100} \quad (17)$$

## 9 Vývoj SW komponent v programovacím prostředí LabVIEW pro firemní testovací SW framework

### 9.1 Požadavky na GUI

- **Hlavní obrazovka** umožňuje:
  - spuštění/zastavení stroje,
  - vyprázdnění podavače a pásu,
  - zobrazení snímku a výsledku posledního vadného kusu ze zvolené kamery,
  - zobrazení základních statistik stroje,
  - vstup do dalších částí aplikace a její ukončení.
- **Servisní panel** umožňuje ovládat veškeré výstupy DIO karty s FPGA a zobrazovat stavy všech vstupů.
- **Konfigurační část** aplikace umožňuje:
  - volbu konfiguračního setu dle typu součástky,
  - nastavení zpoždění triggeru pro každou kameru, zpoždění vzduchového ventilu s tryskou a délku jeho pulzu,
  - nastavení limitů jednotlivých parametrů součástek, možnost přepnout jednotky z pixelů na mm,
  - upravovat parametry inspekce a vyhledávání v obrazu (síla hran, ROI atp.),
  - nasnímat buffer obrázků a z něj automaticky navrhnout limity parametrů,
  - zobrazení detailnějších statistik.
- Veškeré obrazovky mají viditelné pouze ty ovládací prvky, které jsou dostupné aktuálně přihlášenému uživateli na základě jeho práv.

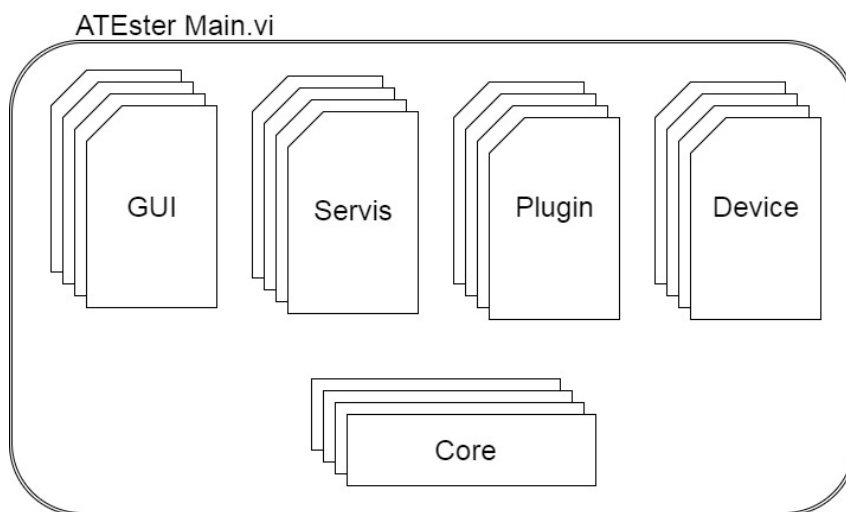
### 9.2 Implementace GUI

Vzhledem k charakteru programování v LabVIEW, kde má každý blokový diagram (tj. kód) automaticky přiřazen svůj čelní panel (tj. uživatelské rozhraní) a k vlastnostem užitého SW frameworku, kde některé SW komponenty mají čelní panel viditelný, vznikalo GUI souběžně s funkcionalitou stroje a jednotlivých SW komponent.

Pro GUI byly využívány grafické ovládací prvky v jednotném stylu firemního frameworku. Uspořádání prvků se postupně částečně vyvíjelo i během testování, kdy docházelo k úpravám na základě požadavků a zkušeností zákazníka (změna umístění některých prvků, jejich zvýraznění, barevné odlišení atp.).

### 9.3 Vývoj v SW frameworku

Pro SW testeru byl využit firemní testovací SW framework, který zajišťuje a standardizuje základní funkčnost (např. asynchronní spouštění SW modulů, komunikace mezi nimi, načítání konfiguračních souborů, error handling atp.), nicméně konkrétní funkčnost bylo nutno implementovat pro automatizovaný tester individuálně na míru. SW framework tak umožňuje velkou modularitu. Veškeré SW komponenty, jejichž čelní panel je viditelný uživateli, mají architekturu tzv. producent – konzument, tj. dvě smyčky, z nichž jedna zachytává události na čelním panelu, druhá vyčítá zprávy z fronty a na základě jejich obsahu volá různé části kódu. Komponenty, které nemají viditelný čelní panel, mají ponechánu pouze část „konzument“, tj. vyčítání zpráv z fronty a volání různých částí kódu. Komponenty mezi sebou komunikují za pomoci front. Software byl tvořen dle nároků firemního standardu, ale také dle doporučení a standardů LabVIEW, jako např. dokumentace subVI, vzhled ikon subVI, volné komentáře u důležitých částí kódu, rozměry a vzhled blokového diagramu, směr toku dat atp. [7], [8], [9]



Obr. 29: Firemní testovací framework

Hlavní součástí SW frameworku je „main“, který zajišťuje základní funkcionalitu aplikace a spouští veškeré ostatní komponenty, viz obr. 29. SW framework zahrnuje několik základních typů komponent dle určení:

- *Core* – logické jádro aplikace, řídí chod aplikace v provozním režimu
- *Device* – zařízení, obstarávají komunikaci s periferiemi (kamery)
- *GUI* – uživatelské rozhraní, hlavní GUI je jediným GUI viditelným pro uživatele v provozním režimu aplikace, vedlejší GUI se zobrazují po přechodu do konfiguračního nebo servisního režimu aplikace.
- *Plugin* – asynchronně volané SW součásti s různorodou funkcností (GUI pro konfiguraci, definice jednotlivých testů atp.)

- *Service\_panels* – servisní panely, slouží pro diagnostiku, interakci se vstupy a výstupy atp.



Obr. 30: Užité SW komponenty

V rámci DP byly pro projekt automatizovaného testeru pro optickou kontrolu SMD součástek nově vyvinuty následující SW komponenty:

- *core\_main.vi* – hlavní logické jádro aplikace, které koordinuje funkci všech ostatních komponent.
- *core\_x\_y.vi* – jádro pro obsluhu jedné kamery a vyhodnocení obrazu z ní, kde x je číslo stanice a y „top“ či „side“ dle umístění kamery. Jádra jsou shodná, jelikož ale SW komponenta typu „core“ není uzpůsobena pro reentrantní běh, jsou potřebná jádra ve 4 samostatných souborech s různými názvy.
- *gui\_main.vi* – hlavní uživatelské rozhraní.
- *plugin\_fpga\_config.vi* – konfigurační GUI pro nastavení zpoždění triggeru kamery, pulzu vzduchového ventilu s tryskou a rychlosti motoru.
- *plugin\_config\_core.vi* – konfigurační GUI pro nastavení vyhledávání v obrazu a limitů měřených parametrů.
- *plugin\_statistics.vi* – komponenta počítající průběžné statistiky v provozním režimu, včetně GUI.
- *service\_panels\_8237R.vi* – servisní panel pro přímé ovládání a zobrazení vstupů/výstupů DIO karty s FPGA.
- *img\_buffer.lvclass* – buffer snímků ve formě třídy, umožňující přiřadit při inicializaci paměť, do které lze sekvenčně cyklicky zapisovat snímky.
- *FPGA\_main\_1* – kód LV FPGA, který je po syntéze nahrán na DIO kartu s FPGA pracující s rychlými signály (přítomnostní čidlo, enkodér, trigger pro kameru).
- *FPGA\_main\_2* – kód LV FPGA, který je po syntéze nahrán na DIO kartu s FPGA pracující s ostatními signály (tlačítko, dveře, signálka, motor, vibrační podavač).

Komponenty typu gui, service\_panels, plugin a core byly postaveny na vzorech, které jsou součástí firemního testovacího SW frameworku. Tyto vzory zajišťují, že všechny komponenty se stejného typu budou mít shodné standardní chování v souladu s navrženým frameworkem (např. při přihlášení uživatele, změně jazyku, ukončení aplikace atp., obdoba interface v textových objektově orientovaných programovacích jazycích).

## 9.4 Hlavní jádro – core\_main

Hlavní jádro zajišťuje základní logiku stroje. Vybraná funkčnost, kterou core\_main zajišťuje:

- Načtení globálních proměnných celé aplikace z .ini souboru.
- Načtení a spuštění bitfile pro provozní režim na FPGA, nastavení konfigurovatelných proměnných (např. zpoždění triggeru kamery, zpoždění vzduchového ventilu s tryskou, PWM motoru).
- Zapnutí hlavního vzduchového ventilu po startu aplikace.
- Cyklickou kontrolu stavu z FPGA. Tato kontrola zahrnuje:
  - inkrementaci proměnné watchdogu,
  - kontrolu stisknutého tlačítka STOP,
  - kontrolu tlaku,
  - kontrolu otevřených dveří.

V případě, že je v kterémkoliv z kroků kontroly zjištěn nepovolený stav (např. zapnutý hlavní ventil, ale nedostatečný tlak; spuštěná inspekce, ale otevřené dveře), je tester okamžitě zastaven a chyba oznámena uživateli. Vyčítání probíhá se 100ms periodou.

- Start a stop provozního režimu stroje (tj. spuštění/vypnutí podavače, motoru).
- Nastavení stavu na signalizačním majáku.
- Kontrolu, zda neuběhl příliš dlouhý čas od poslední kontroly součástky, je-li stroj spuštěn. Slouží pro ošetření např. prázdného vibračního podavače nebo zaseknuté součástky.
- Kontrolu, zda strojem neprojelo příliš mnoho součástek, které byly vyhodnoceny jako vadné (je-li tato funkčnost povolena). Jelikož je vzhledem k charakteru výroby nepravděpodobné, že by strojem projelo za sebou např. 20 vadných kusů, jsou tímto ošetřeny jinak nezachytitelné chybové stavy (např. obsluha vybrala konfigurační soubor pro jiný typ součástek, než strojem projíždí, porucha osvětlovače...).
- Vyprázdnění pásu a celého stroje. Vyprázdnění pásu je automaticky ukončeno po jednom kompletním oběhu pásu, vyprázdnění celého stroje po uplynutí časového limitu nebo do zásahu obsluhy.



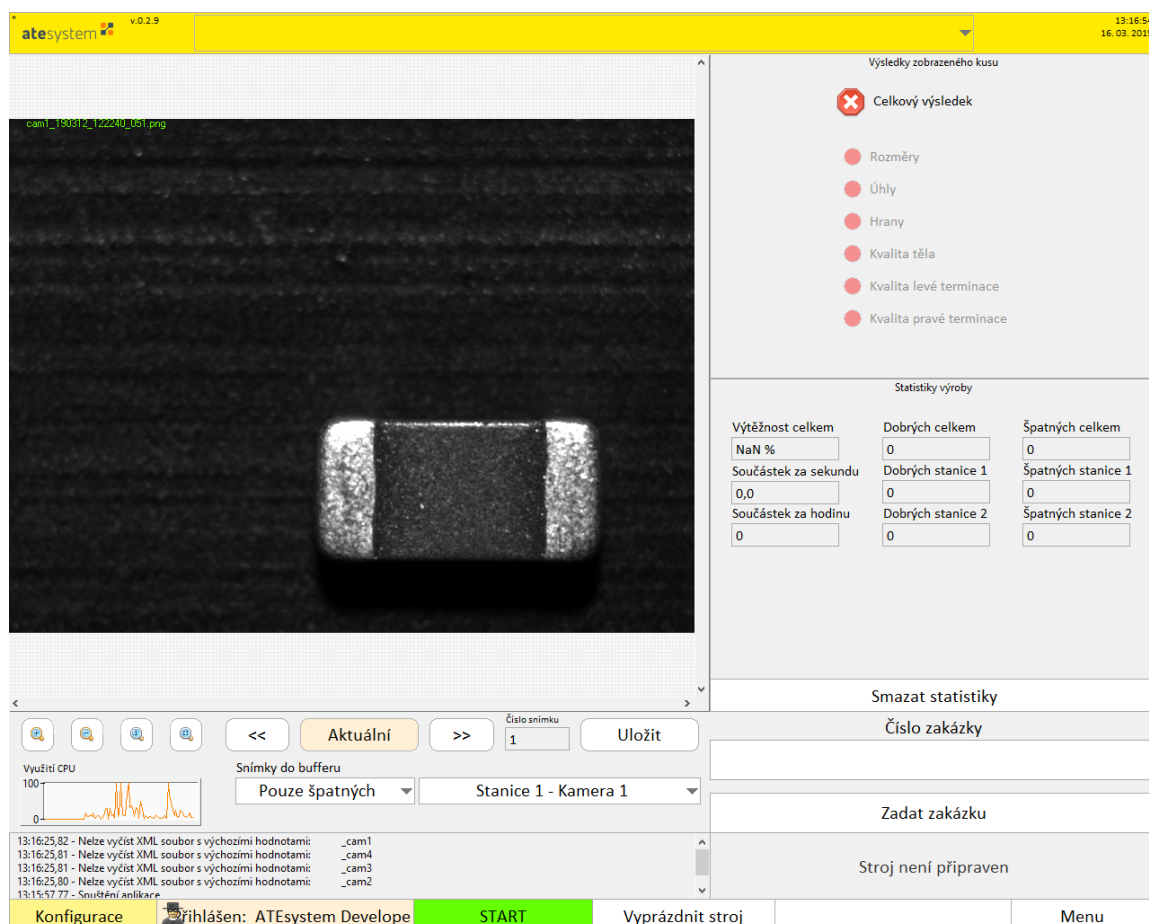
## 9.5 Jádru pro kameru – core\_x\_y

Pro každou ze 4 kamer existuje samostatné jádro. „X“ v názvu je číslo stanice a „y“ „top“ či „side“ dle umístění kamery. Toto jádro má za úkol:

- Sejmout snímek z kamery a vložit jej do bufferu pro zpracování.
- Vyčíst ID aktuálně vyhodnocované součástky z FPGA.
- Vyhodnotit obraz z kamery, výsledek včetně snímku uložit do bufferu zpracovaných snímků a zapsat výsledek včetně příslušného ID na FPGA.

## 9.6 Hlavní uživatelské rozhraní – gui\_main

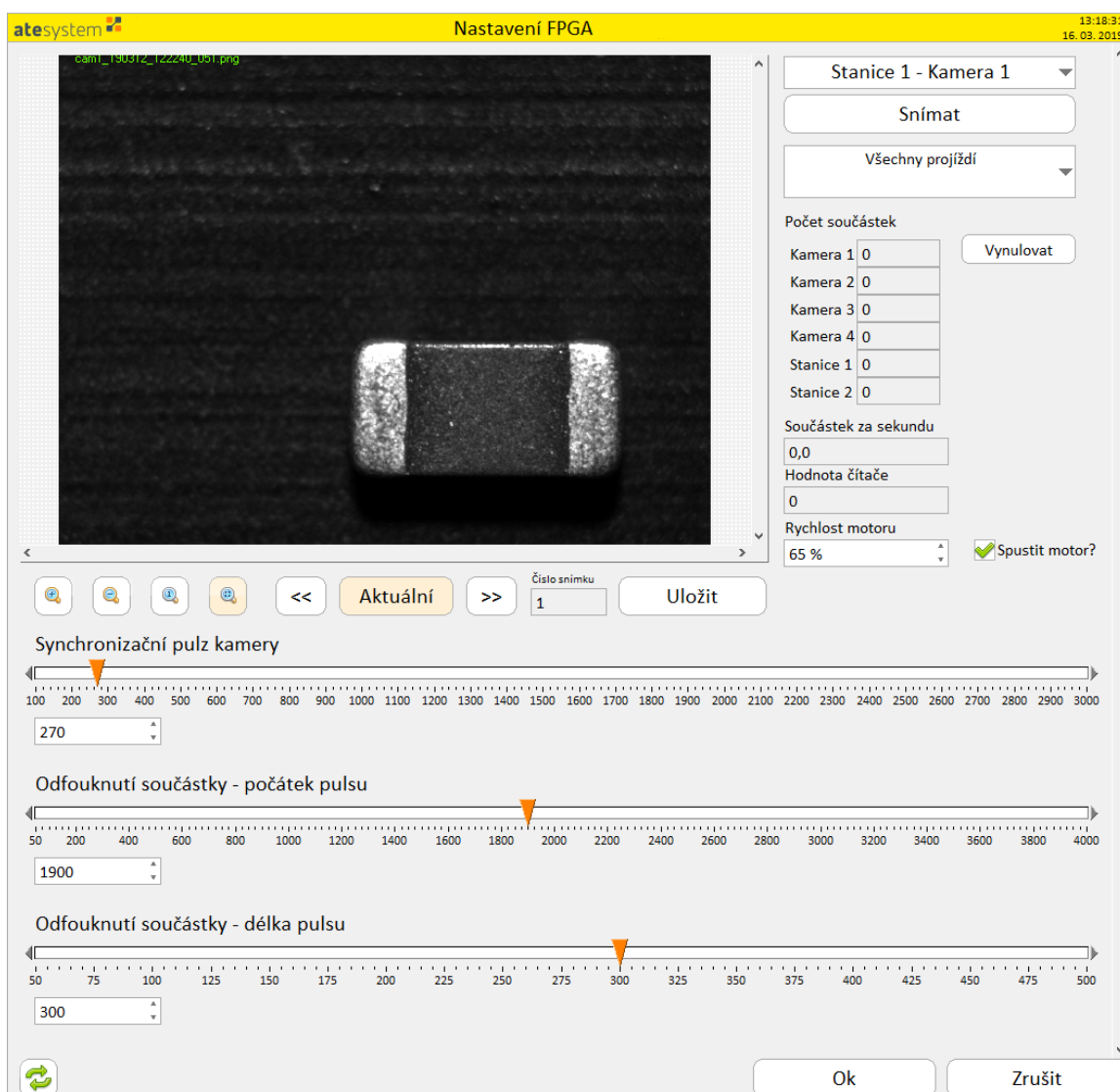
Hlavní uživatelské rozhraní zobrazuje uživateli/operátorovi nejdůležitější provozní informace stroje, protože je jediné, které je viditelné v provozním režimu stroje. Zobrazuje poslední vadnou součástku ze zvolené kamery. V pravé části jsou informace o vyhodnocení kusu, statistiky výroby a informace o zakázce. V dolní části se nachází informační panel se stavem stroje a posledními událostmi stroje. Uživateli je umožněno při zastavené inspekci procházet buffer se snímky.



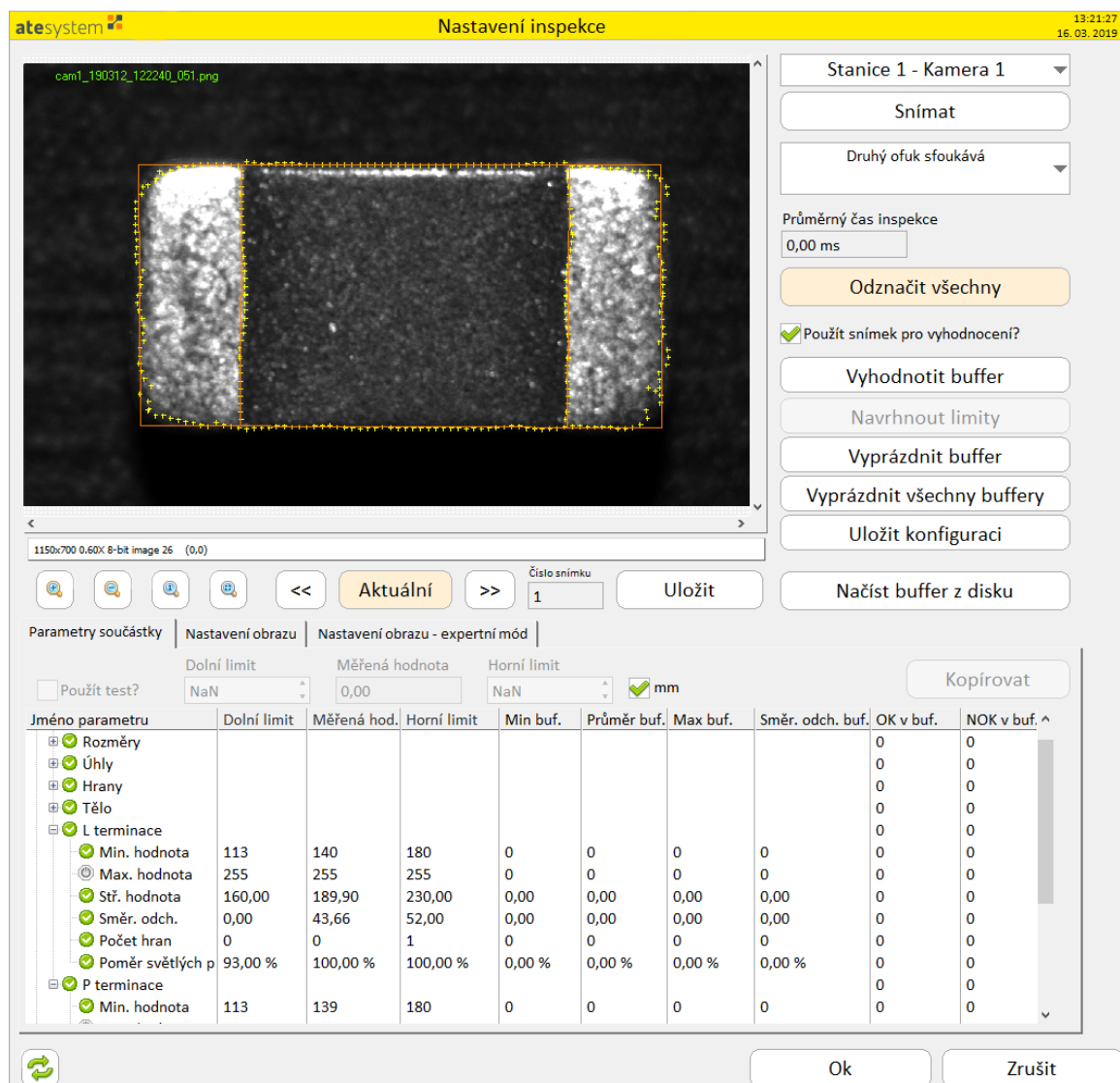
Obr. 31: Hlavní uživatelské rozhraní

## 9.7 Konfigurační GUI pro FPGA – plugin\_fpga\_config

Umožňuje uživateli nastavit parametry FPGA, tj. rychlost motoru (střída PWM), zpoždění triggeru kamery, počátek a délku pulsu vzduchového ventilu. Zároveň zobrazuje aktuální hodnotu čítače, frekvenci součástek za sekundu a jednoduchou statistiku (počet nasnímaných součástek každou kamerou a počet součástek z přítomnostního čidla). Slouží pro ladění průjezdu součástek strojem – pro ladění přítomnostního čidla, pro přesné nastavení snímání kamerou, pro přesné nastavení vzduchového pulzu a pro ladění rychlosti pásu a vibračního podavače vzhledem k žádané frekvenci. Jako zpětná vazba jsou zobrazeny snímky z kamery, aby uživatel ihned viděl, jaký mají změny parametrů efekt. V případě změny některého z FPGA parametrů jsou nové hodnoty zapsány do konfigurační FGV (součást užitého SW frameworku) a za pomoci fronty jsou notifikovány jádra, která upravené hodnoty zapisují do FPGA.



Obr. 32: Konfigurační GUI pro FPGA



Obr. 33: Konfigurační GUI pro nastavení limitů

## 9.8 Konfigurační GUI pro nastavení limitů – plugin\_config\_core

Umožňuje uživateli nastavit limity jednotlivých měřených parametrů v obraze a měnit vyhledávací parametry v obraze. Plugin načítá limity z konfigurační FGV a zobrazuje je uživateli přehledně v tabulce v záložce *Parametry součástky*. Umožňuje načíst celý buffer (z disku nebo z kamery) snímků a ten hromadně vyhodnocovat. Měřené parametry jsou taktéž zobrazeny v tabulce.

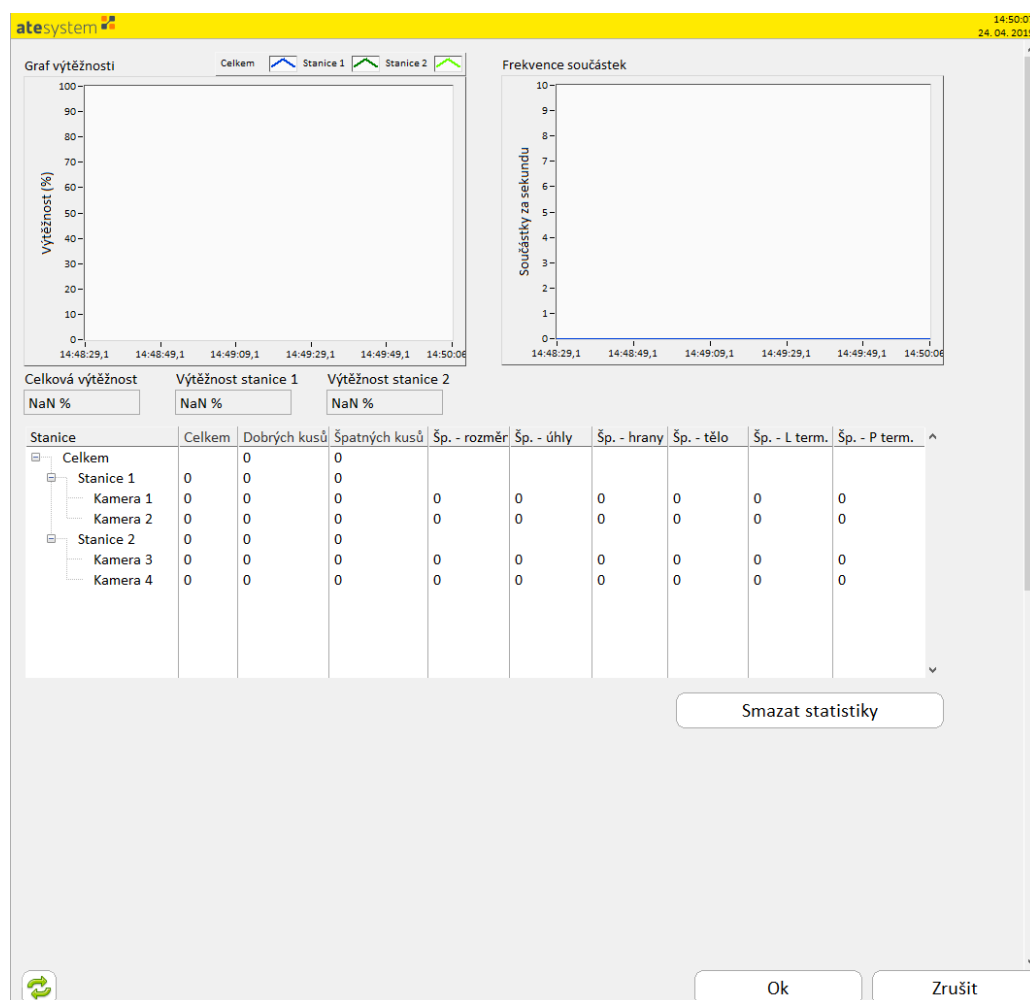
V záložce *Nastavení obrazu* jsou k dispozici nejdůležitější parametry vyhledávání v obraze. Jedná se zejména o nastavení hran a ořezu obrazu.

Záložka *Nastavení obrazu – expertní mód* umožňuje konfigurovat veškeré dostupné parametry vyhledávání v obraze. Je zpřístupněna pouze uživateli s příslušnými právy. Umožňuje zobrazit všechny dílčí snímky, které jsou využity pro zpracování. Jelikož je nesprávnou změnou parametrů

možno prodloužit dobu inspekce a tím snížit maximální takt stroje, je uživateli při hromadném zpracování bufferu zobrazena informativní hodnota průměrné doby inspekce v ms.

## 9.9 Statistiky výroby – plugin\_statistics

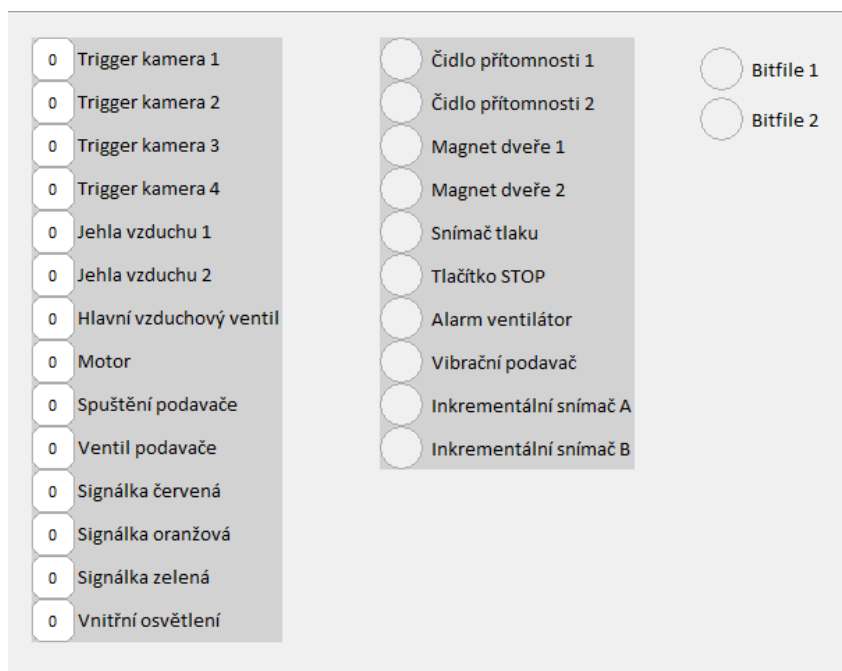
Tento plugin periodicky počítá aktuální statistiky výroby, viz obr. 34. S každým výsledkem z jádra kamery jsou informace o dané součástce (kamera, výsledek, výsledek v kategoriích) uloženy do pole, na jehož základě jsou každou sekundu počítány aktuální statistiky – výtěžnost celková a na konkrétní stanici, celkový počet bezvadných a vadných kusů, frekvence součástek. Uživateli jsou zobrazeny grafy s průběhem za posledních cca 30 minut zobrazující frekvenci a výtěžnost, dále dílčí statistiky v tabulce. Uživatel má možnost statistiku smazat. Statistiky jsou navázány na konkrétní číslo zakázky, které uživatel zadá na hlavní obrazovce. Po ukončení aplikace se aktuální statistiky ukládají do textového souboru, aby byly k dispozici po opětovném spuštění stroje, nedochází ale k jejich archivaci např. do databáze.



Obr. 34: GUI pro zobrazení statistik

## 9.10 Servisní panel – service\_panels\_8237R.vi

Servisní panel je určen pro testování vstupů a výstupů DIO karet s FPGA. Protože s implementovanou provozní funkcionalitou FPGA by bylo náročné přímo přistupovat ke vstupům a výstupům, je při vstupu do servisního panelu nahrán servisní bitfile, který žádanou funkcionalitu zajišťuje. Také to ale znamená, že v tomto módu nejsou aktivní bezpečnostní funkce stroje, tj. tlačítko STOP nevypíná hlavní ventil, otevřené dveře nezastavují motor atp.



Obr. 35: Servisní panel aplikace

## 9.11 Buffer snímků – img\_buffer.lvclass

Jedná se o LabVIEW třídu, která slouží pro ukládání snímků. V aplikaci má 8 instancí – 4 (1 pro každé jádro, o velikosti 5 snímků) mezi smyčkou pro snímání obrazu z kamery a smyčkou pro zpracování obrazu a další 4 pro ukládání vyhodnocených snímků (velikost 200 snímků) a jejich zobrazení uživateli. Při inicializaci dojde k jednorázovému přidělení paměti pro celý buffer, práce s bufferem je tak při běhu aplikace velmi rychlá. Ke každému snímku je možno uložit libovolná data ve formě datového typu variant. Buffer je možno procházet sekvenčně cyklicky, prochází se jen ta část pole, kde byly předem uloženy nějaké snímky (tj. pokud se zaplní jen 50 snímků z 200snímkového bufferu, prochází se cyklicky pouze oněch 50 snímků). V případně zaplnění celého bufferu se začínají snímky přepisovat od nejstaršího. Seznam metod a dat třídy na obr. 36.

img_buffer.lvclass
- Image Array: IMAQ Image[] - Data Array : Variant[] - Image out : IMAQ Image - total_no_images : I32 - array_start : I32 - array_end : I32 - get_image_index : I32 - image_index : I32 - ID : String
+ clear + delete + get_current_image + get_newest_img + get_next_img + get_oldest_img_and_delete + get_previous_img + get_reference + get_total_buffer_size + get_total_no_images + init + insert_data + insert_img

Obr. 36: Diagram třídy img\_buffer

Data:

- Image Array – pole se snímky,
- Data Array – pole s daty ve formě variantu,
- Image out – reference, přes kterou je předáván snímek z pole mimo třídu,
- total\_no\_images – celkový počet uložených snímků v bufferu,
- array\_start – index pole, kde je uložen nejstarší snímek,
- array\_end – index pole, kde je uložen nejnovější snímek,
- get\_image\_index – index pole, ze kterého bylo čteno naposledy,
- image\_index – index snímku v bufferu (1 – nejnovější, zvyšuje se ke starším).

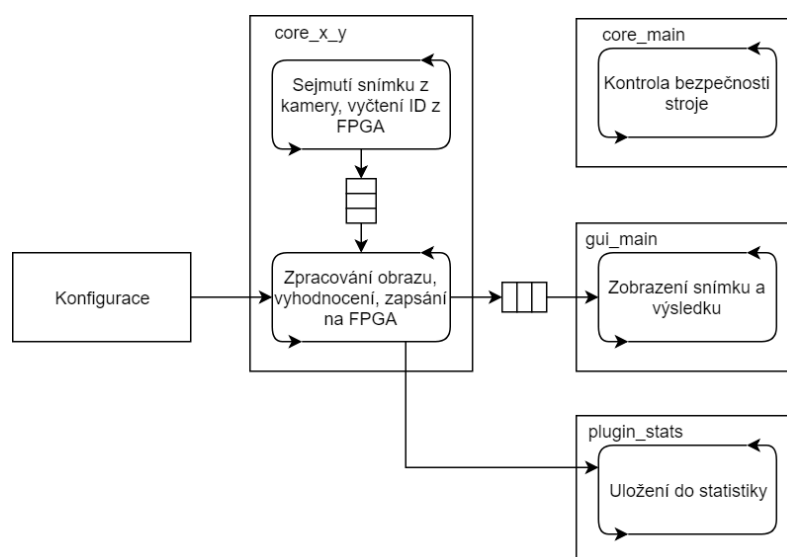
Metody:

- clear – vymaže snímky z bufferu, ale nedealokuje paměť,
- delete – dealokuje paměť bufferu,
- get\_current\_image – vrátí naposledy čtený snímek,
- get\_newest\_image – vrátí nejnovější snímek,
- get\_next\_img – vrátí následující snímek,
- get\_oldest\_img\_and\_delete – vrátí nejstarší snímek a vymaže jej z bufferu,
- get\_previous\_img – vrátí předchozí snímek,
- get\_reference – vrátí referenci na instanci třídy. Důležité, aby bylo možno mít shodnou instanci třídy ve více komponentech SW, protože třída se v LabVIEW ve výchozím stavu předává hodnotou, nikoliv referencí,

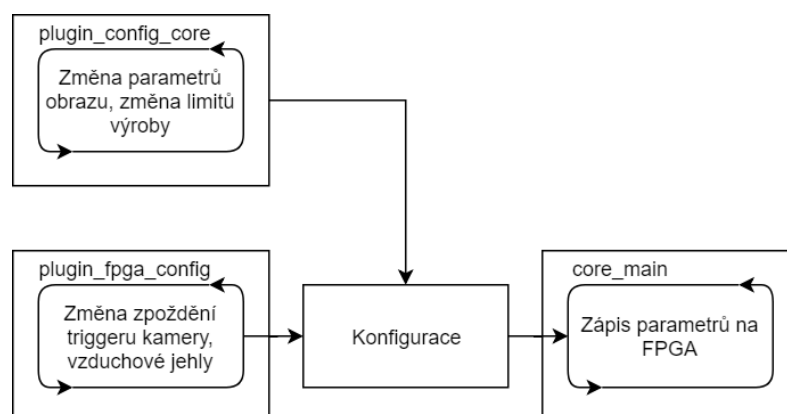
- `get_total_buffer_size` – vrátí alokovanou velikost bufferu (počet snímků),
- `get_total_no_images` – vrátí počet vložených snímků v bufferu,
- `init` – alokuje paměť pro zadaný počet snímků,
- `insert_data` – vkládá pouze k aktuálnímu snímku pouze data,
- `insert_img` – vkládá snímek i s daty.

## 9.12 Popis vztahů jednotlivých komponent aplikace v runtime

Dynamické chování aplikace je zachyceno na diagramech níže. Obr. 37 zachycuje chování v inspekčním módu, obr. 38 zachycuje chování v konfiguračním módu. Komponenta *Konfigurace* je FGV z firemního frameworku. Zbylé komponenty jsou součástí řešení DP, znázorněny jsou i jednotlivé smyčky, shrnuta jejich stručná funkce a tok dat mezi nimi.



Obr. 37: Komunikace mezi jednotlivými SW komponentami v inspekčním módu



Obr. 38: Komunikace mezi jednotlivými SW komponentami v konfiguračním módu

## 10 Nasazení a odladění vyvinutých komponent v průmyslové praxi

### 10.1 Nastavení PC, instalace SW

PC v průmyslovém provedení byl individuálně sestaveno tak, aby splňoval požadavky kladené touto aplikací. Byl vybaven základní deskou a CPU Intel řady i5 starší generace, která ještě má ovladačovou podporu ve Win7. Důvodem potřeby užití Win7 byl fakt, že karty NI PCIe-8237R mají starší FPGA čip Xilinx Spartan 6, pro které jsou k dispozici kompilační nástroje pouze pro Win7. Na PC bylo nainstalováno vývojové prostředí LabVIEW s Vision Acquisition Software, Vision Development Module a FPGA Module. Byla ověřena funkce všech SW a HW komponent PC.

### 10.2 Testování

Testování vytvářených SW komponent probíhalo postupně v souladu s tím, jak byly dostupné oživené jednotlivé části elektrického zapojení stroje. Vývoj a testování probíhaly přibližně v tomto pořadí:

- sekvence zpracování obrazu,
- FPGA část aplikace (algoritmus pro triggerování kamer a sfoukávání součástek z pásu),
- jádra pro sejmutí a zpracování obrazu,
- konfigurace FPGA,
- konfigurace vizuální inspekce a konfigurace limitů a parametrů
- statistika.

Vzhledem k charakteru programování v LabVIEW bylo při vývoji každé komponenty, která je viditelná uživateli, vytvořeno i uživatelské rozhraní. V průběhu vývoje se objevovaly nové požadavky na již vyvinuté komponenty, proto bylo potřeba provést zpětně dílčí úpravy.

Vývoj, testování a validace jednotlivých komponent probíhala několika způsoby dle charakteru komponent.

Komponenty, které nevyžadovaly ke svému běhu koncové PC a externí HW, jsem testoval na vývojovém PC (např. sekvence zpracování obrazu, konfigurace vizuální inspekce). Na koncovém PC jsem poté prováděl pouze benchmark sekvence zpracování obrazu s ohledem na dodržení požadovaného taktu kontroly 6–8 součástek za sekundu. Při benchmarku bylo sledováno využití CPU a čas inspekce, který se musel udržet pod cca 60 ms. Tyto požadavky se postupnou úpravou sekvence zpracování obrazu podařilo splnit.

Část aplikace běžící na FPGA jsem nejprve testoval v simulačním režimu pod Windows na vývojovém PC (FPGA Desktop Execution Node, který je součástí balíku vývojových nástrojů od



NI). Takto jsem mohl otestovat pouze navržený algoritmus, nikoliv však např. možné problémy s časováním fyzických signálů, proto jsem poté prováděl i testování na cílovém PC s FPGA.

Komponenty vyžadující pro svůj běh koncové PC a externí HW, jehož simulace by byla obtížná a časově náročná, jsem testoval a případně i vyvíjel přímo na koncovém PC. Komponenty účastnící se běhu aplikace v provozním režimu (nikoliv konfiguračním či servisním) jsem též podroboval benchmarkům, při kterých jsem opět sledoval zejména časovou náročnost běhu aplikace a využití CPU, opět s ohledem na dodržení požadovaného taktu kontroly 6–8 součástek za sekundu.

Před předáním zákazníkovi byly vytvořeny vzorové konfigurace pro řady SMD součástek, které se měly podrobovat optické kontrole na automatizovaném testeru. Před převozem proběhla kontrolní návštěva, při které si zákazník otestoval, zda stav a funkčnost stroje odpovídají jeho požadavkům.

Po předání zákazníkovi probíhaly rozsáhlé testy, kdy byly do stroje vloženy SMD komponenty různých řad, různé kvality a následně bylo provedeno porovnání vytríděných součástek s nároky oddělení kvality. Během testování se objevily některé problémy – např. součástky jedné z výrobních sérií, které nebyly k dispozici při vývoji, měly podobnou barvu, jako gumový pás, na kterém projíždějí strojem, což způsobovalo problémy při detekci součástky. Objevily se požadavky na úpravu algoritmu ale také na změnu, zjednodušení, přidání či ubrání funkcionality a uživatelského rozhraní. Všechny tyto problémy a požadavky se v průběhu řešení dařilo postupně řešit.

## 11 Závěr

Výsledkem této práce je software, který je nasazen v průmyslové výrobě na automatizovaném testeru pro optickou kontrolu a třídění SMD součástek. Práce zahrnuje výběr hardware pro snímání obrazu, návrh a implementaci software pro řízení, zpracování obrazu, statistické zpracování dat a uživatelské rozhraní v grafickém programovacím jazyce LabVIEW. Nedílnou součástí je nasazení, testování a ladění software ve výrobě.

Hardwarové komponenty pro snímání obrazu, tj. kamera, objektiv, osvětlovač a kontrolér, byly vybrány s ohledem na požadavky na automatizovaný tester (velikost kontrolovaných součástek, takt kontroly). Součástí výběru bylo i testování vybraných komponent na vzorcích SMD součástek.

Řízení stroje je částečně realizováno dvěma DIO kartami s FPGA. Vzhledem k univerzálnosti programovacího prostředí LabVIEW bylo možno v něm implementovat i část pro FPGA. FPGA zajišťuje rychlé čítání a přesné časování, které je nutné pro správné pořízení snímků SMD součástek a pro jejich třídění.

Nadřazená aplikace na PC je postavena na modulárním SW frameworku, implementovaném také v LabVIEW, který umožňuje základní funkčnost aplikace (spouštění jednotlivých komponent, načítání konfiguračních souborů atp.), nicméně bylo nutné přidat do SW frameworku funkčnost specifickou pro tuto konkrétní aplikaci. V praktické části jsou uvedeny nejdůležitější části návrhu, realizace a implementace, včetně seznamu vyvinutých SW komponent a jejich funkce.

Důležitou fází bylo testování a ladění software. Testování částečně probíhalo během vývoje, a to formou testů jednotlivých SW komponent, větších celků, FPGA části a nakonec kompletního SW testeru. Po dokončení probíhalo také testování přímo v průmyslové výrobě. Během ní byly odstraněny drobné chyby v SW, upraven vzhled některých GUI na základě poznatků zákazníka z testování a také byly doladěny konfigurační parametry obrazové inspekce.

Realizovaný software je v současné době nasazen na automatizovaném testeru přímo v průmyslové výrobě. Po počáteční fázi ladění parametrů algoritmu vizuální kontroly, která byla vzhledem ke komplexnosti celé úlohy velice důležitá, se podařilo stroj nastavit tak, aby byly splněny všechny požadavky výroby a kontroly u zákazníka. Během testování se objevily také nové požadavky, drobné nedostatky a chyby v SW, ty se však postupně podařilo úspěšně vyřešit. Stroj tak je v současnosti kompletně předán do užívání zákazníkovi, plní svou funkci spolehlivě a správně vzhledem ke kladeným požadavkům zákazníka.

Pro realizaci jsem využil velké množství znalostí nabytých během studia – od volby hardware pro kamerovou inspekci, přes návrh software, až po realizaci návrhu aplikace v grafickém programovacím jazyce LabVIEW, a to jak pro PC, tak pro FPGA. Diplomová práce mne obohatila o spoustu nových zkušeností spojených s realizací komplexního řešení SW pro automatizovaný tester do průmyslové výroby, a to zejména z oblasti návrhu poměrně rozsáhlého SW, jeho následné implementace a testování.

## Literatura

- [1] BILÍK, Petr a kolektiv, 2012. *Měřicí systémy a jejich programování: učební text a návody do cvičení*. Ostrava.
- [2] GigE Vision Main Page - AIA Vision Standards, 2019. *AIA / Vision Online - Vision Systems* [online]. [cit. 2019-02-17]. Dostupné z: <https://www.visiononline.org/vision-standards.cfm>
- [3] *Camera Link* [online], 2019. [cit. 2019-02-17]. Dostupné z: <http://kamery.atesystem.cz/produkty/komunikace/camera-link/>
- [4] *CCD vs. CMOS: Facts and Fiction* [online], 2001, 3 [cit. 2019-02-17]. Dostupné z: <https://www2.cs.duke.edu/courses/cps274/compsci527/cps274/fall11/papers/Littwiller01.pdf>
- [5] *CMOS vs CCD sensors. Which is better? Comparison of the Image Pixel Sensor Architectures* [online]. [cit. 2019-02-24]. Dostupné z: [http://meroli.web.cern.ch/lecture\\_cmos\\_vs\\_ccd\\_pixel\\_sensor.html](http://meroli.web.cern.ch/lecture_cmos_vs_ccd_pixel_sensor.html)
- [6] KWON, Kye-Si a Steven READY. *Practical guide to machine vision software: an introduction with LabVIEW*. Weinheim: Wiley-VCH, 2015, 300 s. ISBN 978-3-527-33756-9.
- [7] VLACH, Jaroslav, Josef HAVLÍČEK a Martin VLACH. *Začínáme s LabVIEW*. Praha: BEN - technická literatura, 2008. ISBN 978-80-7300-245-9.
- [8] BRESS, Thomas J. *Effective LabVIEW programming*. 1st ed. Allendale, NJ, 2013 701 s. ISBN 978-193-4891-087.
- [9] BITTER, Rick, Taqi MOHIUDDIN a Matt NAWROCKI. *LabVIEW advanced programming techniques*. 2nd ed. Boca Raton: CRC Press, c2007, 499 s. ISBN 978-0849333255.
- [10] *About - OpenCV library* [online]. [cit. 2019-02-23]. Dostupné z: <https://opencv.org/about.html>
- [11] ŠONKA, Milan, Václav HLAVÁČ a Roger BOYLE. *Image processing, analysis, and machine vision*. 4th edition. United States of America: Cengage Learning, 2015, 872 s. International Edition. ISBN 978-1-133-59369-0.
- [12] *IMAQ Count Objects 2 VI - NI Vision 2017 for LabVIEW Help - National Instruments* [online]. [cit. 2019-03-16]. Dostupné z: [http://zone.ni.com/reference/en-XX/help/370281AD-01/imaqvision/imaq\\_count\\_objects\\_2/](http://zone.ni.com/reference/en-XX/help/370281AD-01/imaqvision/imaq_count_objects_2/)
- [13] *IMAQ Rake 3 VI - NI Vision 2017 for LabVIEW Help - National Instruments* [online]. [cit. 2019-03-16]. Dostupné z: [http://zone.ni.com/reference/en-XX/help/370281AD-01/imaqvision/imaq\\_rake\\_3/](http://zone.ni.com/reference/en-XX/help/370281AD-01/imaqvision/imaq_rake_3/)

- [14] *HLV3-22RD-4S / CCS INC.* [online]. [cit. 2019-03-02]. Dostupné z:  
<https://www.ccs-grp.com/products/model/3917>
- [15] *Basler ace acA2040-35gm - Area Scan Camera* [online]. [cit. 2019-03-02]. Dostupné z:  
<https://www.baslerweb.com/en/products/cameras/area-scan-cameras/ace/aca2040-35gm/>
- [16] *VS-MC1-40 - Vital Vision Technology Pte Ltd* [online]. [cit. 2019-03-02]. Dostupné z:  
<https://vitalvisiontechnology.com/machine-vision-lenses/low-distortion-macro-lens/vs-mc-fixed-magnification-models/vs-mc1-40/>

## Přílohy

### Elektronická příloha

Adresář elektronické přílohy obsahuje několik podadresářů:

- *gui* – obsahuje snímky z GUI aplikace v plném rozlišení.
- *vady* – obsahuje plné rozlišení snímků kontrolovaných vad součástek. Typ vady je popsán ve jménu souboru.
- *zdrojovy\_kod\_img\_buffer* – obsahuje LabVIEW zdrojový kód třídy `img_buffer.lvclass`.
- *zdrojovy\_kod\_fpga* – obsahuje zdrojový kód LabVIEW FPGA pro kartu pro zpracování rychlých signálů.